Uniwersytet Warszawski
Wydział Matematyki, Informatyki i Mechaniki

Karol Kański
Nr albumu: 248362

Optymalizacja systemu rekomendacyjnego na podstawie bazy USOS

Praca magisterska
na kierunku MATEMATYKA
w zakresie MATEMATYKI STOSOWANEJ

Praca wykonana pod kierunkiem
dra inż. Przemysława Biecka
Instytut Matematyki Stosowanej i Mechaniki
Zakład Statystyki Matematycznej

Wrzesień 2012

Oświadczenie kierującego pracą

Potwierdzam, że niniejsza praca została przygotowana pod moim kierunkiem i kwalifikuje się do przedstawienia jej w postępowaniu o nadanie tytułu zawodowego.

Data Podpis kierującego pracą

Oświadczenie autora (autorów) pracy

Świadom odpowiedzialności prawnej oświadczam, że niniejsza praca dyplomowa została napisana przeze mnie samodzielnie i nie zawiera treści uzyskanych w sposób niezgodny z obowiązującymi przepisami.

Oświadczam również, że przedstawiona praca nie była wcześniej przedmiotem procedur związanych z uzyskaniem tytułu zawodowego w wyższej uczelni.

Oświadczam ponadto, że niniejsza wersja pracy jest identyczna z załączoną wersją elektroniczną.

Data Podpis autora (autorów) pracy

Streszczenie

W pracy przedstawione jest wprowadzenie do systemów rekomendacyjnych ze szczególnym uwzględnieniem systemów rekomendacyjnych opartych o sąsiedztwo oraz wykorzystywanych w nich algorytmów. Opisano m.in. algorytmy z rodziny najbliższych sąsiadów, algorytm slope one i metody rozkładu macierzy, w których nie wszystkie pola są określone. Oprócz teorii zaprezentowane są wyniki symulacji systemu rekomendacyjnego na danych pochodzących z bazy Uniwersyteckiego Systemu Obsługi Studiów (USOS).

Słowa kluczowe

systemy rekomendacyjne, metoda najbliższych sąsiadów, algorytm slope one, stochastyczna metoda najszybszego spadku, rozkład macierzy

Dziedzina pracy (kody wg programu Socrates-Erasmus)

11.2 Statystyka

Klasyfikacja tematyczna

62-07, 62P99

Tytuł pracy w języku angielskim

A recommendation system for the USOS database

Spis treści

Spis rysunków

Spis tablic

Wprowadzenie

Systemy rekomendacyjne, to stosunkowo młoda, ale dynamicznie rozwijająca się gałąź nauki z pogranicza statystki, data miningu i uczenia maszynowego. Badania w tej dziedzinie zostały wymuszone przez rozwój Internetu i wynikający z niego natłok informacji. Systemy rekomendacyjne mogą być szczególnie przydatne użytkownikom, którzy mają słabe rozeznanie w danym temacie lub też nie potrafią sobie poradzić z olbrzymią liczbą danych, a mimo tego chcieliby znaleźć przedmioty, które mogą okazać się dla nich interesujące. Oczywiście nie ma nic za darmo i w przypadku systemów rekomendacyjnych wkładem, który trzeba wnieść, są dane zawierające odpowiednie informacje. W zależności od typu systemu rekomendacyjnego do takich informacji mogą należeć: opisy przedmiotów, historia zamówień lub informacja na temat tego, jak użytkownik ocenia wybrane przedmioty.

Okazuje się, że tego typu informacje można znaleźć w bazie danych Uniwersyteckiego Systemu Obsługi Studiów (USOS). Z punktu widzenia systemu rekomendacyjnego interesujące mogą być: opisy kursów, to na jakie kursy uczęszczał dany student, jakie dostał z nich oceny i jak sam je oceniał (w niektórych wersjach systemu USOS studenci mogą wypełnić ankiety dotyczące kursów, w których uczestniczyli). Co więcej, w dzisiejszych czasach uczelnia wyższa oferuje studentom szeroką ofertę programową, z której mogą oni wybierać kursy. Powyższe czynniki sprawiają, że szkoła wyższa jest dobrym miejscem do wprowadzenia systemu rekomendacyjnego, który sugerowałby studentom mogące ich zainteresować kursy. Celem niniejszej pracy jest właśnie zastosowanie metod używanych w systemach rekomendacyjnych do sugerowania studentom kursów i ocena jakości takich rekomendacji. Jedyną różnicą w stosunku do większości systemów rekomendacyjnych jest to, że nie będziemy korzystać z ocen z ankiet, gdzie studenci faktycznie oceniają dany kurs, a z ocen, które uzyskali oni z danego kursu. Studentom byłyby więc polecane przedmioty, z których mają szansę uzyskać najwyższą ocenę.

Praca składa się z 4 rozdziałów i dwóch dodatków. Rozdział 1 skład się z dwóch części: pierwsza zawiera wprowadzenie do systemów rekomendacyjnych, natomiast w drugiej znajduje się szczegółowy opis systemów rekomendacyjnych opartych o sąsiedztwo, które są głównym tematem tej pracy. W rozdziale 2 można znaleźć teorię dotyczącą metod obliczeniowych użytych podczas wykonywania analiz do tej pracy. Rozdział 3 zawiera szczegółowy opis analiz wykonanych na danych z systemu USOS. Można tam znaleźć informację na temat tego, jakie dane były dostępne, jak je przygotowywano, jakie analizy zostały wykonane i wreszcie krótki opis uzyskanych wyników. Ostatni rozdział stanowi

podsumowanie pracy. W dodatku A zamieszczono kompletne wyniki ocen poszczególnych metod, a w dodatku B można znaleźć spis zawartości płyty DVD dołączonej do niniejszej pracy.

Rozdział 1

Systemy rekomendacyjne

Znaczna część użytkowników Internetu miała w ten lub inny sposób do czynienia z systemami rekomendacyjnymi. W większości sklepów internetowych (lub portali z wiadomościami) jesteśmy informowani o najpopularniejszych obecnie artykułach (wiadomościach), a przeglądając szczegóły konkretnego przedmiotu znajdujemy sekcję "ci, którzy kupili ten przedmiot, kupili również" ("ci, którzy przeczytali ten artykuł, przeczytali również"). Są to przykłady tzw. rekomendacji niespersonalizowanych, które dla każdego użytkownika są takie same. Takie rekomendacje często są skuteczne, bo skoro coś już trafiło w dużą liczbę gustów, to pewnie spodoba się też większości użytkowników, którzy jeszcze nie natrafili na dany przedmiot. Okazuje się jednak, że wyższą skuteczność rekomendacji można osiągnąć, generując je w spersonalizowany sposób, tzn. sugerując danemu użytkownikowi przedmioty, które spodobają się akurat jemu, a niekoniecznie większości. Tego typu rekomendacje można znaleźć np. w serwisach takich, jak: Amazon.com, Google (warto spróbować wyszukać to samo hasło z różnych komputerów i porównać kolejność wyników), czy YouTube.

Powyższy akapit powinien dać ogólny pogląd na temat tego, co rozumiemy przez system rekomendacyjny, warto jednak przytoczyć trochę bardziej formalną definicję. W [28] system rekomendacyjny definiuje się jako oprogramowanie i techniki pozwalające na sugerowanie użytkownikom przedmiotów w taki sposób, jakby sugestia pochodziła od innych użytkowników. W niniejszej pracy nieco większy nacisk będzie położony na "techniki", aczkolwiek większość opisanych metod została również zaimplementowana (w języku R), a ich skuteczność zmierzona. Warto w tym momencie wspomnieć, że w rozwój systemów rekomendacyjnych miało wkład wiele różnorodnych dziedzin nauki, takich jak np.: uczenie maszynowe, data mining, interakcje człowiek-komputer (ang. *human-computer interaction*), rachunek prawdopodobieństwa, statystyka, czy pozyskiwanie informacji (ang. *information retrieval*).

Systemy rekomendujące w sposób spersonalizowany operują przede wszystkim na dwóch pojęciach: przedmiotu i użytkownika. Przedmioty to byty, które można rekomendować użytkownikom. W przypadku niniejszej pracy będą to kursy prowadzone na uczelni. Przedmioty i użytkownicy są łączeni za pomocą rankingów (ocen), które dany użytkownik wystawił danemu przedmiotowi, przy czym nie dla każdej pary użytkownik-

przedmiot ranking musi istnieć (w większości systemów rekomendacyjnych rankingi istnieją dla zaledwie kilku procent par). Ponadto wyróżnia się rankingi bezpośrednie i pośrednie. Z tymi pierwszymi mamy do czynienia, kiedy użytkownik faktycznie ocenił przedmiot, natomiast te drugie to np. fakt, że użytkownik zakupił dany przedmiot lub chociaż oglądał jego opis. Czasem system rekomendacyjny ma jeszcze dostęp do pewnych dodatkowych informacji takich, jak: data wystawienia rankingu, opis przedmiotu, pewne informację nt. użytkownika (wiek, płeć, miejsce zamieszkania itd.). W zależności od tego, jakiego typu jest to system rekomendacyjny (patrz rozdział 1.2) wybiera się część dostępnych informacji, na podstawie których tworzy się model użytkownika i ostatecznie generuje rekomendacje.

Dalsza część tego rozdziału rozpoczyna się od opisu trudności stających przed systemami rekomendacyjnymi, przy czym mamy tu na myśli problemy istotne z punktu widzenia algorytmów tworzących rekomendacje (a nie np. bezpieczeństwa działającego systemu, czy tego w jaki sposób prezentować listę sugerowanych przedmiotów). Dalej znajduje się opis trzech najbardziej popularnych typów systemów rekomendacyjnych. Rozdział kończy się pogłębioną dyskusją nt. systemów rekomendacyjnych opartych o sąsiedztwo, które są głównym tematem niniejszej pracy.

1.1. Wyzwania systemów rekomendacyjnych

W zasadzie od samego początku badań nad systemami rekomendacyjnymi największym wyzwaniem jest pokonanie dwóch przeciwstawnych problemów:

- rzadkich danych;
- skalowalności dla danych dużych rozmiarów.

W pierwszym przypadku prawdziwy problem polega na tym, żeby znaleźć sposób prognozowania w sytuacji, w której mamy dostępną małą ilość danych. W typowym systemie rekomendacyjnym przeciętny użytkownik ocenił zaledwie kilka przedmiotów, co powoduje, że przewidywanie tego jak oceniłby pozostałe jest trudne. Jednym z pomysłów na rozwiązanie tego problemu jest wykorzystanie dodatkowych informacji o użytkowniku (płeć, wiek, pochodzenie). Podobne podejście stosuje się również w przypadku tzw. problemu "zimnego startu", czyli nowych użytkowników i przedmiotów, o których nie mamy zupełnie żadnych danych. Najczęściej jednak takie dodatkowe informacje nie są dostępne i nie można tego problemu rozwiązać w ten sposób. Inny sposobem na radzenie sobie z rzadkimi danymi jest ich uzupełnienie. Problemem jest oczywiście to, w jaki sposób to zrobić. Ponieważ w zasadzie nie ma dobrej odpowiedzi na to pytanie, to takie podejście stosuje się rzadko, zwłaszcza, że prowadzi ono do drugiego z wymienionych na wstępie problemów, czyli braku skalowalności. W niektórych obecnie używanych systemach rekomendacyjnych mamy miliony użytkowników i dziesiątki tysięcy przedmiotów. Posiadanie rankingów dla każdej pary użytkownik-przedmiot sprawiłoby, że ciężko byłoby przeprowadzać jakiekolwiek obliczenia biorąc pod uwagę wszystkie dane. W przypadkach, gdy mamy do czynienia z taką ilością danych, to jednym z rozwiązań jest zastosowanie którejś z metod redukcji wymiaru (PCA, SVD).

Dla zilustrowania skali powyższych problemów przytoczymy dane na temat rozmiaru i postaci danych w systemie rekomendacyjnym witryny Netflix [24]. Dane te zostały udostępnione w związku z ogłoszonym przez tą witrynę konkursem na system rekomendacyjny [25]. Konkurs polegał na tym, by poprawić wynik (rozumiany jako pierwiastek błędu średniokwadratowego prognoz rankingów) dotychczasowego systemu rekomendacyjnego witryny o przynajmniej 10%, a główną nagrodą było 1.000.000$. Wynik ten udało się osiągnąć dopiero po trzech latach od rozpoczęcia konkursu. Sam konkurs spowodował też bardzo dynamiczny rozwój badań nad systemami rekomendacyjnymi. W konkursie udostępniono uczestnikom zbiór treningowy, który składał się ze 100 milionów rankingów. Na ten zbiór złożyły się oceny 480189 użytkowników przyznane 17770 przedmiotom. Oznacza to, że znane było zaledwie 1.17% wszystkich rankingów (rzadkie dane), a gdyby wszystkie rankingi były znane, to mielibyśmy do czynienia ze zbiorem zawierającym około 10^{10} rekordów (problemy ze skalowalnością).

1.2. Typy systemów rekomendacyjnych

W chwili obecnej można wyróżnić trzy najważniejsze typy systemów rekomendacyjnych:

- oparte o sąsiedztwo;
- oparte o treść;
- oparte o wiedzę.

To jaki system rekomendacyjny najlepiej sprawdzi się w danym zastosowaniu zależy najczęściej od tego, jaki informacje będą dla niego dostępne. Dosyć istotna jest też charakterystyka dziedziny, w której system ma rekomendować (jak często użytkownicy potrzebują rekomendacji, czy często zdarza się, że trzeba coś rekomendować nowemu użytkownikowi). Ponieważ każdy z wymienionych wyżej typów systemów ma swoje wady, to dosyć często stosuje się również rozwiązania hybrydowe. W niniejszej pracy zajmujemy się systemami opartymi o sąsiedztwo, ale żeby dać pełen obraz badań toczących się na polu systemów rekomendacyjnych, kolejne sekcje zawierają krótki opis każdego z najważniejszych typów systemów rekomendacyjnych.

1.2.1. Oparte o sąsiedztwo

Systemy rekomendacyjne oparte o sąsiedztwo (ang. *colaborative filtering*) są dziś najpopularniejszym i najszerzej stosowanym typem systemów rekomendacyjnych. Badania nad nimi trwają od przeszło 15 lat, co sprawia, że są dość dobrze zrozumiane i opisane. Podstawowy pomysł opiera się tutaj na założeniu, że jeśli dwóch użytkowników podejmowało w przeszłości podobne decyzje (kupowali te same przedmioty, czytali te same artykuły, podobnie oceniali filmy), to w przyszłości ich preferencje też będą zbieżne. Przykładowo, jeżeli zbiór filmów wypożyczonych przez użytkownika A w dużej mierze pokrywa się z tymi wypożyczanymi przez B oraz istnieje jakiś film K, który był wypożyczony przez A i nie wypożyczony dotąd przez B, to sensownie jest zasugerować B

wypożyczenie filmu K. W rzeczywistości, aby zarekomendować coś danemu użytkownikowi wyszukuje się użytkowników najbardziej do niego podobnych, a następnie sprawdza, które z lubianych przez nich przedmiotów można zarekomendować rozważanemu użytkownikowi.

Sąsiedztwo może odnosić się także do sąsiedztwa przedmiotów. Użytkownikowi rekomendujemy przedmioty najbardziej podobne do tych, które ocenił najwyżej. Więcej szczegółów znajduje się w rozdziale 1.3. Do najważniejszych zalet systemów rekomendacyjnych opartych o sąsiedztwo należą:

- nie potrzeba żadnych dodatkowych danych poza historią ocen;
- łatwa implementacja;
- potrafią odkrywać ciekawe zależności.

Nie są one też niestety pozbawione wad:

- problem "zimnego startu" dla nowych użytkowników i przedmiotów;
- wymagają tego, żeby użytkownicy wystawili przedmiotom jakieś oceny.

1.2.2. Oparte o treść

Systemy rekomendacyjne oparte o treść (ang. *content-based*) są również dosyć popularne. Ich działanie opiera się na założeniu, że dostępne są jakieś dodatkowe informacje na temat przedmiotów. W przypadku filmów mogą to być np.: opis, reżyser, gatunek, lista aktorów, kraj i rok produkcji. Podobnie jak w przypadku systemów opartych o sąsiedztwo również tutaj wyszukujemy przedmioty podobne do przedmiotów dotąd wybieranych przez użytkownika. Tym razem jednak podobieństwo jest obliczane na podstawie dostępnej treści, a nie historii ocen. W przypadku filmów za podobne można uznać te, które są wyreżyserowane przez tą samą osobę, mają ten sam gatunek lub których listy aktorów się pokrywają. Bardzo istotnym elementem obliczania podobieństwa jest porównywanie opisów przedmiotów. Jest to część większej dziedziny nauki nazywanej pozyskiwaniem informacji. W skrócie polega to na zidentyfikowaniu słów kluczowych w danym opisie (odrzucenie tych, które znajdują się w większości tekstów w danym języku), przyporządkowanie im istotności (może zależeć od ilości wystąpień), a następnie porównanie ze słowami kluczowymi z innego opisu.

Dosyć ciekawą cechą tych systemów jest fakt, że nie muszą być one wykorzystywane do rekomendacji, a mogą służyć użytkownikom do wyszukiwania interesujących ich pozycji. Do zalet tych systemów można zaliczyć następujące:

- użytkownicy nie muszą oceniać przedmiotów;
- nie jest potrzebna duża społeczność;
- nowe przedmioty mogą być od razu rekomendowane;

- potrafią w zrozumiały sposób uzasadnić dlaczego rekomendują ten, a nie inny przedmiot (np. w przypadku filmów, bo pokrywa się reżyser, trzech aktorów i gatunek).

Wady tych systemów są następujące:

- konieczność utrzymywania opisów przedmiotów;
- "zimny start" dla nowych użytkowników;
- rezultaty są raczej przewidywalne – użytkownik nawet bez użycia systemu rekomendacyjnego mógłby się domyślić, że dane przedmioty będą mu się podobać.

1.2.3. Oparte o wiedzę

Systemy rekomendacyjne oparte o wiedzę (ang. *knowledge-based*) spotyka się rzadziej od pozostałych dwóch opisanych rodzajów. Istnieją jednak zastosowania, w których ten typ systemów sprawdza się najlepiej. Dobrym przykładem może być internetowy sklep z narzędziami ogrodniczymi. W takim sklepie ciężko jest liczyć na istnienie historii zamówień użytkownika, ponieważ taki sprzęt kupuje się rzadko. Przedmioty mogą za to mieć bardziej dokładny i ustrukturyzowany opis. W związku z tym użytkownika, który chce kupić kosiarkę można zapytać o to, jaka jest wielkość jego działki, czy często kosi albo jakiego rodzaju trawę posiada i na podstawie udzielonych odpowiedzi zarekomendować mu produkt, który najlepiej spełni jego oczekiwania. Mówiąc bardziej ogólnie, w takich systemach istnieje funkcja, która ocenia to w jakim stopniu dany przedmiot jest w stanie rozwiązać postawiony przez użytkownika problem. Zaletami takich systemów są:

- użytkownicy nie muszą oceniać przedmiotów, nie potrzeba dużej społeczności, ani historii zamówień – nie ma problemu "zimnego startu";
- wysoka jakość rekomendacji.

Ten typ systemów ma też swoje wady:

- konieczność utrzymywania dokładnych opisów przedmiotów;
- spory wysiłek na tzw. "inżynierię wiedzy", czyli dodawanie i utrzymywanie informacji o przedmiotach w ustrukturyzowany sposób;
- nie uczą się w trakcie działania, więc zawsze będą proponowały te same przedmioty.

1.3. Cechy systemów opartych o sąsiedztwo

W niniejszym rozdziale zostaną dokładniej opisane pojęcia i algorytmy, z których korzystają systemy rekomendacyjne oparte o sąsiedztwo wprowadzone w sekcji 1.2.1. Ponieważ ma to być opis formalny, w sekcji 1.3.1 opisane są założenia i notacja używane w dalszej części pracy. Kolejne sekcje tego rozdziału zawierają już właściwy opis metod używanych w systemach rekomendacyjnych opartych o sąsiedztwo.

1.3.1. Założenia i notacja

W rozdziale 1 wymieniono kilka nazw dla różnych bytów wchodzących w skład systemów rekomendacyjnych, które dotąd były stosowane zamiennie. Od tej pory, w celu uniknięcia nieporozumień będziemy używali następujących określeń: użytkownik, przedmiot i ranking lub ocena. To założenie przestanie być aktualne w rozdziale 3, gdzie będzie mowa o: studencie, kursie i ocenie.

Użytkowników będziemy zazwyczaj oznaczali symbolami u oraz v, natomiast dla przedmiotów rezerwujemy symbole i oraz j. Zarówno użytkowników, jak i przedmioty będziemy zazwyczaj utożsamiali z przypisanym im kodem (identyfikatorem, id). Będziemy też zakładali, że mamy w systemie M użytkowników i N przedmiotów, a kody przypisane użytkownikom i przedmiotom to liczby całkowite ze zbiorów $\{1..M\}$ i $\{1..N\}$ odpowiednio. Ranking przyznany przez użytkownika u przedmiotowi i będzie oznaczany przez r_{ui}, natomiast prognoza tego rankingu, w sytuacji gdyby był on nieznany będzie oznaczana przez $\hat{r}_{ui}$. Rankingi r_{ui} najczęściej należą do jakiegoś podzbioru liczb wymiernych (np. $\{1, 2, 3, 4, 5\}$ lub $\{1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5\}$), natomiast prognoza może być dowolną liczbą rzeczywistą. Średni ranking wystawiony przez użytkownika u to $\bar{r}_u$, a średni ranking przyznany przedmiotowi i to $\bar{r}_i$. W obu przypadkach do średniej liczą się tylko znane rankingi. Wobec powyższych rankingi można ustawić w macierz R wymiaru $M \times N$ zdefiniowaną następująco:

$$R = [r_{ui}]_{ui} \tag{1.1}$$

w której u-ty wiersz oznacza rankingi u-tego użytkownika, a i-ta kolumna rankingi i-tego przedmiotu. Należy przy tym pamiętać, że w macierzy R nie wszystkie pozycje będą zdefiniowane (patrz też sekcja 1.1).

Zbiór dopuszczalny rankingów będzie oznaczany przez $\mathcal{S}$. Ponadto będziemy wyróżniali jeszcze następujące zbiory: $\mathcal{U} = \{1..M\}$ – zbiór użytkowników, $\mathcal{I} = \{1..N\}$ – zbiór przedmiotów, $\mathcal{R}$ – zbiór znanych rankingów, czyli trójek (u, i, r_{ui}). Zbiór przedmiotów ocenionych przez użytkownika u będzie oznaczany przez $\mathcal{I}_u$, a zbiór użytkowników, którzy ocenili przedmiot i to $\mathcal{U}_i$. Przydatne mogą być jeszcze oznaczenia dla przedmiotów ocenionych przez dwóch konkretnych użytkowników $\mathcal{I}_{uv} = \mathcal{I}_u \cap \mathcal{I}_v$ i użytkowników, którzy ocenili dwa konkretne przedmioty $\mathcal{U}_{ji} = \mathcal{U}_j \cap \mathcal{U}_i$. $|\mathcal{R}|$ oznacza moc (liczbę elementów) zbioru $\mathcal{R}$. Inne oznaczenia będą wprowadzane w miarę potrzeby.

1.3.2. Miary podobieństwa

Jednym z najpopularniejszych algorytmów w systemach rekomendacyjnych opartych o sąsiedztwo jest algorytm najbliższych sąsiadów (ang. *nearest neighbor* – NN, patrz sekcja 1.3.3). Dla skuteczności działania tego algorytmu bardzo istotny jest sposób określania podobieństwa (odległości) pomiędzy użytkownikami lub przedmiotami.

Kłam twierdzeniu, że dobór odpowiedniej miary podobieństwa jest jedną z najważniejszych decyzji podczas projektowania systemu rekomendacyjnego zadają wyniki pracy [19], gdzie zmiana miary podobieństwa nie miała większego wpływu dla jakości rekomendacji. Z drugiej strony z pracy [31], w której znajduje się porównanie różnych miar

podobieństwa dla rekomendacji w systemie społecznościowym Orkut, wynika, że miara podobieństwa jednak ma znaczenie i najlepiej sprawdzają się miary kosinusowe (patrz dalej). Jednym z wyników poniższej pracy będzie też odpowiedź na pytanie o znaczenie doboru miary podobieństwa.

W tej sekcji znajduje się opis różnych miary podobieństwa stosowanych w systemach rekomendacyjnych, a także sposoby normalizacji rankingów, które mogą poprawić jakość wyników otrzymywanych za pomocą tych miar. Jeśli nie zaznaczono inaczej użytkownik, bądź przedmiot będą reprezentowani przez wektor rankingów. W miejsce nieznanych rankingów formalnie wpisujemy symbol "NA" i pozycje z tym symbolem nie są brane pod uwagę. Będziemy też zakładać, że wektory należą do przestrzeni $\mathbb{R}^n$, a do ich oznaczania będziemy używać symboli x i y, gdzie $x = (x_1, x_2, ..., x_n)$ oraz $y = (y_1, y_2, ..., y_n)$.

Odległość w l^p

Najprostszą i najbardziej znaną miarą odległości między dwoma punktami jest odległość euklidesowa:

$$d_2(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}. \tag{1.2}$$

Uogólnienie metryki euklidesowej stanowi odległość wektorów w przestrzeni l^p:

$$d_p(x, y) = \left(\sum_{i=1}^{n}(x_i - y_i)^p\right)^{\frac{1}{p}}. \tag{1.3}$$

Metryka l^p przyjmuje wartości z przedziału $[0, +\infty)$ i oczywiście im większa jej wartość tym badane punkty są bardziej odległe i tym samym mniej podobne.

Licząc odległość dwóch użytkowników lub przedmiotów za pomocą metryki l^p pod uwagę brane są jedynie te współrzędne, które są zdefiniowane dla obu wektorów. Uwzględniając ten fakt, odległość dwóch użytkowników możemy zapisać jako:

$$d_p(u, v) = \left(\sum_{i \in \mathcal{I}_{uv}}(r_{ui} - r_{vi})^p\right)^{\frac{1}{p}}. \tag{1.4}$$

Korelacja Pearsona

W przypadku badania odległości użytkowników najbardziej popularną miarą jest tzw. korelacja Pearsona. Definiuje się ją w następujący sposób:

$$Pearson(x, y) = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}, \tag{1.5}$$

co w przypadku porównywania dwóch użytkowników, gdzie bierzemy pod uwagę fakt, że część rankingów nie jest znana, można zapisać następująco:

$$Pearson(u, v) = \frac{\sum_{i \in \mathcal{I}_{uv}}(r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\sum_{i \in \mathcal{I}_{uv}}(r_{ui} - \bar{r}_u)^2 \sum_{i \in \mathcal{I}_{uv}}(r_{vi} - \bar{r}_v)^2}}. \tag{1.6}$$

Korelacja Pearsona przyjmuje wartości z przedziału $[-1, 1]$, przy czym im wartość jest wyższa tym użytkownicy są bardziej podobni.

Kosinus

Podobieństwo dwóch wektorów można również mierzyć za pomocą kosinusa kąta pomiędzy nimi:

$$cos(x, y) = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}}. \tag{1.7}$$

Kosinus kąta pomiędzy wektorami jest najczęściej używany do mierzenia podobieństwa przedmiotów, dlatego możemy zapisać:

$$Cosine(i, j) = \frac{\sum_{u \in \mathcal{U}_{ij}} r_{ui} r_{uj}}{\sqrt{\sum_{u \in \mathcal{U}_{ij}} r_{ui}^2}\sqrt{\sum_{u \in \mathcal{U}_{ij}} r_{uj}^2}}. \tag{1.8}$$

Mierząc podobieństwo przedmiotów za pomocą miary kosinusowej w żaden sposób nie uwzględniamy tego, że dany ranking r_{ui} może być bardzo wysoki (niski) nie dlatego, że użytkownik u bardzo lubi (nie lubi) przedmiot i, a dlatego, że ma on tendencję do oceniania wysoko (nisko) wszystkich przedmiotów. Dlatego też wprowadza się miarę "dostosowany" (ang. *adjusted*) kosinus, która usuwa z oceny przedmiotu efekt związany z użytkownikiem:

$$AdjCosine(i, j) = \frac{\sum_{u \in \mathcal{U}_{ij}} (r_{ui} - \bar{r}_u)(r_{uj} - \bar{r}_u)}{\sqrt{\sum_{u \in \mathcal{U}_{ij}} (r_{ui} - \bar{r}_u)^2}\sqrt{\sum_{u \in \mathcal{U}_{ij}} (r_{uj} - \bar{r}_u)^2}}. \tag{1.9}$$

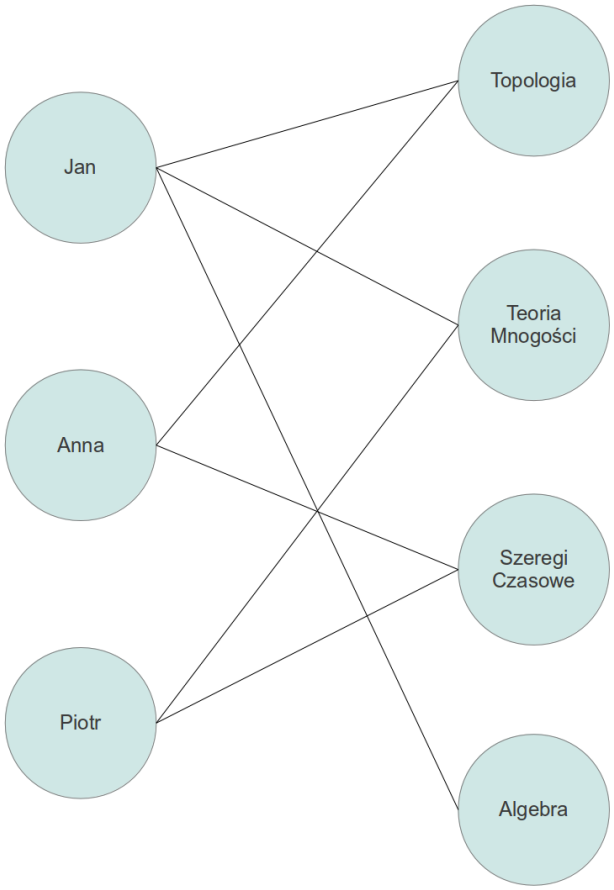
Miary oparte o kosinus przyjmują wartości z przedziału $[-1, 1]$, przy czym im wartość jest wyższa tym przedmioty są bardziej podobne.

Wzór 1.9 jest dosyć podobny do tego definiującego korelację Pearsona (wzór 1.6), dlatego warto podkreślić różnicę między nimi. Licząc podobieństwo przedmiotów za pomocą dostosowanego kosinusa, od rankingu odejmujemy średnią z rankingów użytkownika, który go wystawił. W przypadku korelacji Pearsona dla przedmiotów (użytkowników) od rankingu odejmujemy średnią ze wszystkich rankingów danego przedmiotu (użytkownika).

Miara oparta o grafy

Wadą rozważanych wcześniej miar jest to, że przy liczeniu podobieństwa dwóch użytkowników (przedmiotów) brane są pod uwagę tylko rankingi tych przedmiotów (użytkowników), które ocenili obaj użytkownicy (przedmioty). Dlatego w przypadku niepełnych danych, takich jakie są dostępne w systemach rekomendacyjnych, dla wielu par tak przedmiotów, jak i użytkowników nie uda się określić podobieństwa.

Jednym ze sposobów radzenia sobie z powyższym problemem jest przedstawienie danych w postaci grafu. W najprostszym przypadku zarówno użytkownicy, jak i przedmioty są reprezentowani jako wierzchołki, natomiast każda krawędź ma jeden koniec



Rysunek 1.1: Dane jako graf. Każdy użytkownik i każdy przedmiot odpowiadają jednemu wierzchołkowi grafu, natomiast krawędź pomiędzy wierzchołkami oznacza, że dany użytkownik ocenił dany przedmiot.

reprezentujący użytkownika, jeden reprezentujący przedmiot i oznacza, że dany użytkownik ocenił dany przedmiot (patrz rysunek 1.1). W poprzednio omawianych miarach pod uwagę brane są tylko ścieżki zawierające trzy wierzchołki (np. dwa przedmioty i użytkownik, który ocenił oba), natomiast takie przedstawienie danych pozwala wziąć pod uwagę również ścieżki zawierające większą ilość wierzchołków.

Istnieje wiele metod, które wykorzystują taką reprezentację (patrz np. [7], [9], [21]). W poniższej pracy zostanie wykorzystana metoda, pochodząca z [12], z tą jednak różnicą, że metoda ta posłuży nam do obliczenia podobieństwa między przedmiotami (użytkownikami), które następnie można wykorzystać w algorytmach NN. We wspomnianej pracy autorzy w analogiczny sposób liczą związek pomiędzy przedmiotem i użytkownikiem i na tej podstawie dokonują rekomendacji.

W metodzie tej aby obliczyć podobieństwo między dwoma elementami liczy się liczbę

różnych ścieżek krótszych niż pewne ustalone n pomiędzy wierzchołkami reprezentującymi je. Ponieważ dłuższe ścieżki oznaczają słabszy związek, to liczbę ścieżek długości i bierze się do sumy z czynnikiem α^i dla pewnego $\alpha \in (0, 1)$. Zatem podobieństwo elementów x oraz y można zapisać jako:

$$GraphSim(x, y) = \sum_{i=1}^{n} \alpha^i \cdot paths(x, y, i), \tag{1.10}$$

gdzie $\alpha \in (0, 1)$, n to maksymalna długość ścieżek, które bierzemy pod uwagę, a $paths(x, y, i)$ oznacza liczbę ścieżek pomiędzy wierzchołkami reprezentującymi x oraz y zawierających dokładnie i wierzchołków.

Orginalnie opisywana metoda odnosiła się do dokonywania zamówień – zakładało się, że zamówienie przez klienta danego przedmiotu oznacza pozytywny ranking dla tego przedmiotu. W przypadku rankingów wyrażanych explicite można np. przyjąć, że krawędź pomiędzy użytkownikiem i przedmiotem występuje w grafie tylko i wyłącznie, jeśli ranking dla tego przedmiotu jest wyższy niż średni ranking wystawiany przez danego użytkownika.

Normalizacja

W sekcji dotyczącej miar kosinusowych wspomniano o tym, że tak naprawdę ciężko jest porównywać rankingi przyznane jednemu przedmiotowi przez dwóch różnych użytkowników. Powodem tego jest fakt, że mimo iż wszyscy użytkownicy korzystają z tej samej skali, to niektórzy mają tendencję do przyznawania tylko wysokich rankingów, inni tylko niskich, a jeszcze inni mogą faktycznie wykorzystywać całą skalę. W tej sytuacji nawet jeżeli użytkownik przyznał przedmiotowi ranking powyżej środka skali, to wcale nie oznacza to, że lubi on ten przedmiot (i na odwrót). W celu uczynienia rankingów porównywalnymi stosuje się pewne normalizacje. Normalizacji rankingów można dokonać w dwóch momentach: przed policzeniem podobieństwa i przed policzeniem prognozy. Warto zwrócić uwagę, że normalizowanie w tych dwóch momentach wykonuje się niezależnie (tzn. można wykonać tylko w jednym, w żadnym albo w obu).

Pierwszym możliwym sposobem normalizacji jest centrowanie danych, tzn. odejmowanie od rankingu średniej z rankingów (użytkownika lub przedmiotu, w zależności na co zwracamy uwagę w danym momencie). Następnie do prognozy opartej na scentrowanych rankingach trzeba dodać średnią którą odjęliśmy. Zakładając, że centrujemy względem użytkowników mamy:

$$r'_{ui} = r_{ui} - \bar{r}_u, \tag{1.11}$$

a prognoza to teraz

$$\hat{r}_{ui} = \bar{r}_u + \widehat{r'}_{ui}. \tag{1.12}$$

Kolejna możliwość normalizacji rankingów, bazująca na z-score, uwzględnia też rozrzut rankingów użytkownika. Motywacją dla tej normalizacji jest następujący przykład: mamy dwóch użytkowników, z których jeden zawsze przyznaje ranking 3, a drugi przyznał tyle samo rankingów 1, co 5. Średnie tych użytkowników są takie same, ale ocena

5 wystawiona przez użytkownika pierwszego byłaby czymś niezwykłym i świadczyłaby o jego dużym uznaniu dla danego przedmiotu. Stąd bierze się pomysł, aby oprócz średniej uwzględniać jeszcze odchylenie standardowe rankingów użytkownika. Zakładając, że odchylenie standardowe rankingów użytkownika u jest oznaczona przez σ_u mamy:

$$r''_{ui} = \frac{r_{ui} - \bar{r}_u}{\sigma_u}, \tag{1.13}$$

a prognoza to teraz

$$\hat{r}_{ui} = \bar{r}_u + \sigma_u \cdot \widehat{r''}_{ui}. \tag{1.14}$$

W przypadku normalizacji dokonywanej przed policzeniem podobieństwa warto zwrócić uwagę, że w niektórych miarach normalizacja jest już w jakiś sposób obecna (korelacja Pearsona, "dostosowany" kosinus), albo nie ma sensu (miary grafowe). Normalizacji należy dokonywać ostrożnie, ponieważ:

- jeżeli użytkownik wystawiał jednakowe rankingi, to normalizacja z-score sprawi, że będą one nieokreślone;
- normalizowanie może sprawić, że prognozy rankingów częściej będą miały wartości większe od największego dopuszczalnego rankingu lub mniejsze od dopuszczalnego minimalnego rankingu.

Kurczenie podobieństw

Prawie wszystkie opisane wcześniej miary mają dosyć istotną wadę. Otóż zwracane przez nie podobieństwo nie bierze pod uwagę tego, na podstawie ilu obserwacji je obliczono. Innymi słowy podobieństwo dwóch obiektów, obliczone na podstawie jednego wspólnego rankingu jest traktowane tak samo, jak podobieństwo dwóch innych obiektów pochodzące od 100 wspólnych rankingów, mimo że to drugie jest dużo bardziej wiarygodne. W celu wyeliminowania tego efektu można zastosować tzw. "kurczenie" podobieństw – podobieństwo zaczyna zależeć od liczby rankingów użytych do jego obliczenia. Najczęściej spotykane przekształcenia podobieństw to (na przykładzie podobieństw przedmiotów; kurczenia podobieństw użytkowników dokonuje się analogicznie):

$$sim'(i,j) = \frac{min(|\mathcal{U}_{ji}|, \gamma)}{\gamma} \cdot sim(i,j) \tag{1.15}$$

$$sim''(i,j) = \frac{|\mathcal{U}_{ji}|}{|\mathcal{U}_{ji}| + \beta} \cdot sim(i,j) \tag{1.16}$$

W przypadku równania 1.15 użyto wartości progowej γ, która decyduje o tym, czy należy dokonać kurczenia wag. Takie podejście opisano w [10] i [22], gdzie odkryto, że wartość $\gamma \geqslant 25$ może znacząco poprawić jakość prognoz. Bardziej ciągłe podejście z równania 1.16 zostało opisane w [3].

W tej pracy będziemy badać jedynie wpływ drugiego ze sposobów kurczenia wag na jakość prognoz.

1.3.3. NN

Algorytm NN (ang. *nearest neighbor* – najbliższych sąsiadów) jest jednym z najprostszych, ale też najpopularniejszych algorytmów służących do klasyfikacji obiektów. W orginalnej wersji obiekty treningowe są reprezentowane przez wektory cechy, z których każdy ma przypisaną klasę do której należy. Zadanie polega na przypisaniu klasy obiektowi dla którego znamy jedynie wektor cech. W tym celu oblicza się odległość (lub podobieństwo) klasyfikowanego obiektu do obiektów ze zbioru treningowego. Następnym krokiem jest wybór pewnej liczby obiektów treningowych położonych najbliżej klasyfikowanego obiektu. Obiekt zostaje zakwalifikowany do klasy, która występuje najliczniej wśród wybranych sąsiadów.

W przypadku systemów rekomendacyjnych, gdzie skala ocen jest dyskretna można użyć algorytmu w formie przedstawionej powyżej. Dużo częściej zdarza się jednak, że skalę ocen można traktować jako ciągłą i wtedy sensowniej jest ocenę klasyfikowanego obiektu obliczyć jako średnią ważoną ocen sąsiadów. W tym przypadku należy się zastanowić jak przypisywać wagi sąsiadom. Niezależenie od tego, czy rozwiązujemy zadanie przewidywania, czy klasyfikacji istotne jest też określenie, które obiekty zaliczamy do zbioru najbliższych sąsiadów danego obiektu.

Wybór sąsiadów

Jest rzeczą oczywistą, że wybór odpowiedniego zbioru sąsiadów jest jednym z ważniejszych czynników wpływających na skuteczność rekomendacji. W praktyce spotyka się dwa sposoby kwalifikowania obiektów jako najbliższych sąsiadów:

- **k najbliższych** – jak sama nazwa wskazuje wybieramy k obiektów, które leżą najbliżej klasyfikowanego obiektu. W metodzie tej największym problemem jest dobranie odpowiedniego k, ponieważ zbyt duża jego wartość powoduje, że na rekomendację mają wpływ bardzo mało podobni sąsiedzi, natomiast zbyt mała wartość sprawia, że niektóre przedmioty mogą nigdy nie zostać zarekomendowane.
- **powyżej wartości progowej** – ustalany jest pewien próg podobieństwa i wszyscy sąsiedzi powyżej tego progu są zaliczani do najbliższych sąsiadów. Ponownie zbyt niska wartość progu powoduje pogorszenie rekomendacji poprzez uwzględnienie bardzo mało podobnych obiektów, natomiast zbyt wysoka wartość wpływa na to, że pod uwagę brana jest bardzo mała liczba obiektów (lub wręcz nie ma żadnego obiektu powyżej progu), co oczywiście też źle wpływa na jakość rekomendacji.

Przypisanie wag sąsiadom

Tak jak już wcześniej wspomniano, mając wybrany zbiór najbliższych sąsiadów, ocenę lub klasę dla danego obiektu można obliczać biorąc pod uwagę to jak bardzo każdy z tych sąsiadów jest podobny do rozważanego obiektu. W tym celu każdemu z sąsiadów przypisuje się wagę i prognoza to średnia ważona ocen sąsiadów, a w przypadku klasyfikacji jest to klasa z najwyższą sumą wag. Najczęściej wagi przypisuje się w następujący sposób:

- jeśli mamy daną miarę podobieństwa to jest to po prostu podobieństwo pomiędzy dwoma obiektami;
- jeśli mamy daną miarę odległości to jest to po odwrotność odległości pomiędzy dwoma obiektami.

1.3.4. Slope one

Algorytm slope one został po raz pierwszy zaprezentowany w pracy [20]. Jego idea opiera się na pojęciu "różnicy popularności" – dla każdej pary przedmiotów bada się o ile jeden przedmiot jest bardziej popularny od drugiego. W ten sposób znając ranking przyznany przez danego użytkownika jednemu przedmiotowi można przewidzieć jaki ranking przyznałby drugiemu przedmiotowi. Funkcja prognozująca jest funkcją liniową o nachyleniu jeden (stąd nazwa): $f(x) = x + b$, gdzie x to ranking, który znamy, $f(x)$ to prognoza, a b to różnica popularności pomiędzy przedmiotem, którego ranking znamy, a tym którego ranking przewidujemy. Ponieważ dany użytkownik najczęściej ocenił więcej niż jeden przedmiot, to przewidując ranking danego przedmiotu możemy po prostu uśrednić prognozy pochodzące od wszystkich przedmiotów, które dany użytkownik ocenił. Różne możliwości policzenia "różnicy popularności" i średniej pochodzącej od prognoz sprawiają, że w rodzinie algorytmów slope one mamy trzy schematy: slope one, ważony slope one i bi-polar slope one.

Przedstawimy teraz formalnie w jaki dokładnie sposób liczone są prognozy slope one. "Różnica popularności" pomiędzy przedmiotami, to średnia różnica w rankingach tych przedmiotów, przy czym uwzględniane są tylko rankingi pochodzące od użytkowników, którzy ocenili oba:

$$dev_{ji} = \sum_{u \in \mathcal{U}_{ji}} \frac{r_{uj} - r_{ui}}{|\mathcal{U}_{ji}|} \tag{1.17}$$

Biorąc pod uwagę, że $r_{ui} + dev_{ji}$ jest prognozą dla $\hat{r}_{uj}$ przy znanym r_{ui} oraz fakt, że użytkownik ocenił więcej niż jeden przedmiot otrzymujemy prognozę slope one dla $\hat{r}_{uj}$:

$$\hat{r}_{uj} = \sum_{i \in \mathcal{I}_{uj}} \frac{r_{ui} + dev_{ji}}{|\mathcal{I}_{uj}|}, \tag{1.18}$$

gdzie $\mathcal{I}_{uj} = \{i | i \in \mathcal{I}_u, i \neq j, |\mathcal{U}_{ji}| > 0\}$ to zbiór tych przedmiotów i, które ocenił użytkownik u, a dodatkowo istnieją użytkownicy, którzy ocenili zarazem i oraz j. Przedmiotu j nie zaliczamy do tego zbioru.

Jedną z poważniejszych wad metody slope one jest fakt, że biorąc średnią z prognoz nie uwzględnia się tego na ilu obserwacjach dane przewidywanie jest oparte. Intuicyjnie, jeżeli $|\mathcal{U}_{ji}| = 10$ oraz $|\mathcal{U}_{jk}| = 1000$ to prognoza dla j oparta o k będzie lepsza niż ta oparta o i. W związku z tym można zdefiniować algorytm ważony slope one, który będzie uwzględniał tą informację:

$$\hat{r}_{uj} = \sum_{i \in \mathcal{I}_u \setminus \{j\}} \frac{(r_{ui} + dev_{ji}) c_{ji}}{\sum_{i \in \mathcal{I}_u \setminus \{j\}} c_{ji}}, \tag{1.19}$$

gdzie $c_{ji} = |\mathcal{U}_{ji}|$.

Kolejnym usprawnieniem jest potraktowanie osobno przedmiotów, które użytkownik lubił i tych, których nie lubił. Dzielimy zbiór przedmiotów ocenionych przez danego użytkownika u na dwa zbiory $\mathcal{I}_u^{like} = \{i | i \in \mathcal{I}_u, r_{ui} \geqslant \bar{r}_u\}$ oraz $\mathcal{I}_u^{dislike} = \{i | i \in \mathcal{I}_u, r_{ui} < \bar{r}_u\}$. Ponadto dla każdej pary przedmiotów wybieramy zbiór użytkowników, którzy oba te przedmioty lubili oraz zbiór użytkowników, którzy obu tych przedmiotów nie lubili: $\mathcal{U}_{ij}^{like} = \{u | i, j \in \mathcal{I}_u^{like}\}$, $\mathcal{U}_{ij}^{dislike} = \{u | i, j \in \mathcal{I}_u^{dislike}\}$. Teraz "różnicę popularności" możemy policzyć dla każdych tych zbiorów osobno:

$$dev_{ji}^{like} = \sum_{u \in \mathcal{U}_{ij}^{like}} \frac{r_{uj} - r_{ui}}{|\mathcal{U}_{ij}^{like}|}, \tag{1.20}$$

$$dev_{ji}^{dislike} = \sum_{u \in \mathcal{U}_{ij}^{dislike}} \frac{r_{uj} - r_{ui}}{|\mathcal{U}_{ij}^{dislike}|}. \tag{1.21}$$

Prognoza dla j oparta o i to $p_{ji}^{like} = u_i + dev_{ji}^{like}$ lub $p_{ji}^{disike} = u_i + dev_{ji}^{dislike}$ w zależności od tego, czy i należy do $\mathcal{I}_u^{like}$, czy do $\mathcal{I}_u^{dislike}$. Ostatecznie przewidywanie algorytmu bipolar slope one dla $\hat{r}_{ui}$ to:

$$\hat{r}_{ui} = \frac{\sum_{i \in \mathcal{I}_u^{like} \setminus \{j\}} p_{ji}^{like} c_{ji}^{like} + \sum_{i \in \mathcal{I}_u^{dislike} \setminus \{j\}} p_{ji}^{dislike} c_{ji}^{dislike}}{\sum_{i \in \mathcal{I}_u^{like} \setminus \{j\}} c_{ji}^{like} + \sum_{i \in \mathcal{I}_u^{dislike} \setminus \{j\}} c_{ji}^{dislike}}, \tag{1.22}$$

gdzie $c_{ji}^{like} = |\mathcal{U}_{ji}^{like}|$ oraz $c_{ji}^{dislike} = |\mathcal{U}_{ji}^{dislike}|$.

1.3.5. Metody bazujące na modelu

Algorytmy opisane w poprzednich sekcjach zaliczały się do tzw. "metod bazujących na pamięci" (ang. *memory-based*). Metody te tworzą prognozy dla użytkowników bazując na ich wcześniejszych rankingach – zazwyczaj jest to średnia ważona tych poprzednich rankingów. W niniejszej sekcji opiszemy pewne algorytmy z innej rodziny, zwanej "metodami bazującymi na modelu" (ang. *model-based*). Algorytmy z tej klasy próbują tworzyć modele opisujące zachowania użytkowników i cechy przedmiotów, a następnie używać tych modeli do generowania prognoz. Do najdokładniej zbadanych algorytmów opartych o modele należą: LDA (latent Dirichlet allocation, [4]), klasteryzacja Bayesowska [5], pLSI (probabilistic latent semantic indexing, [11]), MMF (Multiple Multiplicative Factor models, [23]) oraz MDP (Markov Decision process, [30]). W tej pracy opiszemy i zbadamy jednak podejście zaproponowane w pracy [29] i dalej rozwijane chociażby podczas konkursu Netflix'a (patrz np. [3], [8], [17], [26]).

Metoda zostanie opisana na przykładzie danych, do których orginalnie ją zastosowano tzn. filmów i wypożyczających je użytkowników. Dane pochodzące z bazy USOS są nieco inne, dlatego pewnych kroków orginalnej metody nie da się do nich zastosować. Pomysły w nich zawarte są jednak bardzo ciekawe i dlatego część z nich zostanie opisana w poniższej sekcji (nie będą to jednak wszystkie kroki pierwotnej wersji).

Wprowadzenie

Modele, o których wspomniano w poprzedniej sekcji, to nic innego jak tylko formuły (funkcje) podające przepis na obliczenie nieznanego rankingu. W formułach tych występują nieznane parametry, które opisują pewne cechy użytkowników i przedmiotów oraz interakcje między nimi. Nieznane parametry są wyznaczane w fazie uczenia modelu, która polega zminimalizowaniu pewnej funkcji łączącej dostępne dane (obserwacje) z parametrami występującymi w modelu. W minimalizowanej funkcji występują najczęściej dwa główne składniki – pierwszy odpowiada za to, żeby model zgadzał się z obserwacjami, natomiast drugi, tzw. czynnik regularyzacyjny ma za zadanie zapobiegać przeuczeniu modelu. Przeuczenie modelu polega na tym, że jest on za bardzo dopasowany do danych treningowych i przez to źle radzi sobie z rzeczywistymi danymi. Regularyzacja polega na karaniu za wielkość parametrów, przy czym kara jest proporcjonalna do metaparametrów oznaczanych przez $\lambda_1, \lambda_2, \lambda_3, ...$

Powyższy opis może wprowadzać wrażenie, że modelowanie jest skomplikowanym procesem, jednak modele przedstawiane w kolejnych sekcjach z pewnością wiele wyjaśnią. W tym rozdziale na opis modeli będzie składała się formuła pozwalająca na wyznaczenie rankingu, funkcja użyta do uczenia modelu oraz wyjaśnienie za co odpowiadają parametry w formule. Opis metod obliczeniowych służących do uczenia modelu (zminimalizowania funkcji) znajduje się w rozdziale 2.2.

Parametry bazowe

Modele w systemach rekomendacyjnych mają przede wszystkim wyłapywać związki pomiędzy użytkownikami i przedmiotami. Zazwyczaj jednak duża część sygnału pochodzi od cech użytkowników (przedmiotów), które nie zależą od przedmiotów (użytkowników). Niektórzy użytkownicy mogą mieć tendencję do przyznawania wysokich ocen, inni wręcz przeciwnie (patrz też rozdział 1.3.2). Część filmów jest tak słaba, że większość użytkowników oceni je poniżej średniej, inne są wybitne, więc będą miały prawie same wysokie oceny. Parametry bazowe (ang. *baseline predictors*) zwane też obciążeniami (ang. *biases*) mają za zadanie wyłapać tego typu zależności. W modelu bazowym nieznany ranking r_{ui} jest przewidywany przez parametr b_{ui} obliczany w następujący sposób:

$$b_{ui} = \mu + b_u + b_i, \tag{1.23}$$

gdzie μ jest średnią oceną dla całego zbioru treningowego, b_u oraz b_i oznaczają zaobserwowane odchylenia od średniej dla użytkownika u i przedmiotu i odpowiednio.

Wyznaczenie parametrów b_u i b_i odbywa się poprzez zminimalizowanie następującej funkcji:

$$\min_{b_*} \sum_{(u,i,r_{ui})\in\mathcal{R}} (r_{ui} - \mu - b_u - b_i)^2 + \lambda_1 \left(\sum_{u\in\mathcal{U}} b_u^2 + \sum_{i\in\mathcal{I}} b_i^2 \right). \tag{1.24}$$

Tak jak wspomniano w poprzedniej sekcji mamy tu dwa główne składniki: $\sum_{(u,i,r_{ui})\in\mathcal{R}}(r_{ui} - \mu - b_u - b_i)^2$ kontroluje jakość dopasowania parametrów b_u i b_i do danych, natomiast

$\lambda_1(\sum_{u\in\mathcal{U}} b_u^2 + \sum_{i\in\mathcal{I}} b_i^2)$ chroni przed przeuczeniem modelu. Metoda pozwalająca na znalezienie minimum powyższej funkcji jest opisana w rozdziale 2.2.

SVD

Rozkład SVD macierzy (ang. *singular value decomposition* – rozkład na wartości szczególne) jest techniką bardzo popularną w statystyce i przetwarzaniu informacji. Jej głównymi zastosowaniami są redukcja wymiaru macierzy oraz bardziej interesujące z naszego punktu widzenia odkrywanie struktury ukrytych czynników (patrz [6]). Rozkład SVD macierzy A to przedstawienie jej w postaci:

$$A = U\Sigma V^T, \tag{1.25}$$

gdzie U i V to macierze ortonormalne, a Σ to macierz diagonalna postaci $\Sigma = diag(\sigma_i)$, przy czym σ_i to wartości szczególne macierzy.

Problemem z orginalnym rozkładem SVD jest to, że jest on niezdefiniowany, jeśli macierz A jest niekompletna. Początkowe próby rozwiązania tego problemu (patrz [29] oraz [15]) polegały na uzupełnianiu macierzy, to jednak powodowało problemy związane z olbrzymią liczbą danych, a poza tym mogło niekorzystnie wpływać na wyniki. Nowocześniejsze rozwiązania tego problemu biorą pod uwagę jedynie dostępne rankingi (patrz [3], [8], [17], [26]). W niniejszej pracy przedstawimy właśnie rozkład macierzy rankingów R, który będzie uwzględniał jedynie znane rankingi.

W prezentowanym podejściu macierz rankingów R rozmiaru $N \times M$ jest rozkładana na macierze P i Q wymiarów odpowiednio $M \times f$ oraz $N \times f$, gdzie f to wymiar przestrzeni ukrytych cech:

$$R \approx PQ^T. \tag{1.26}$$

Oczywiście dokładny rozkład może być niemożliwy, dlatego równość jest przybliżona. Chcemy zminimalizować pewną funkcję (w tym wypadku daną wzorem 1.28) macierzy $R - PQ^T$, biorącą pod uwagę jedynie te miejsca, gdzie macierz R jest określona.

Model uwzględniający rozkład macierzy posiada też parametry bazowe zdefiniowane w poprzednim rozdziale, a zatem formuła wygląda następująco:

$$b_{ui} = \mu + b_u + b_i + q_i p_u, \tag{1.27}$$

gdzie q_i oznacza i-ty wiersz macierzy Q, p_u u-ty wiersz macierzy P, a $q_i p_u$ ich iloczyn skalarny. Macierze P i Q można traktować jako rzuty użytkowników i przedmiotów na f-wymiarową przestrzeń ukrytych cech, gdzie kolejne wiersze oznaczają użytkowników lub przedmioty, a kolejne pozycje w wierszu to, jaka jest zgodność danego użytkownika lub przedmiotu z daną cechą. Dopasowanie tego modelu następuje poprzez zminimalizowanie następującej funkcji:

$$\min_{b_*,q_*,p_*} \sum_{(u,i,r_{ui})\in\mathcal{R}} (r_{ui} - \mu - b_u - b_i - q_i p_u)^2 + \lambda_2 \left(\sum_{u\in\mathcal{U}} b_u^2 + \sum_{i\in\mathcal{I}} b_i^2 + \sum_{u\in\mathcal{U}} \|p_u\|^2 + \sum_{i\in\mathcal{I}} \|q_i\|^2 \right). \tag{1.28}$$

SVD++

Model nazwany SVD++ (zaproponowany w [17] i [28]) różni się od poprzedniego tym, że oprócz przyznanych rankingów bierze pod uwagę też rankingi pośrednie. W tym konkretnie przypadku rankingiem pośrednim jest sam fakt, że użytkownik zdecydował się w ogóle ocenić dany film. Użytkownik jest więc teraz reprezentowany przez wektor cech, jak w przypadku zwykłego SVD, a także zbiór filmów, które ocenił. Na potrzeby modelowania obecności w zbiorze ocenionych filmów z każdym z nich związany jest dodatkowy wektor $y_i \in R^f$, wtedy model wygląda następująco:

$$b_{ui} = \mu + b_u + b_i + q_i \left(p_u + |\mathcal{R}_u|^{-0.5} \sum_{j\in R_u} y_j \right). \tag{1.29}$$

Jak łatwo się domyślić minimalizowana funkcja to teraz

$$\min_{b_*,q_*,p_*,y_*} \sum_{(u,i,r_{ui})\in\mathcal{R}} \left(r_{ui} - \mu - b_u - b_i - q_i(p_u + |\mathcal{R}_u|^{-0.5} \sum_{j\in R_u} y_j) \right)^2 + \lambda_3 \left(\sum_{u\in\mathcal{U}} b_u^2 + \sum_{i\in\mathcal{I}} b_i^2 + \sum_{u\in\mathcal{U}} \|p_u\|^2 + \sum_{i\in\mathcal{I}} \|q_i\|^2 + \sum_{i\in\mathcal{I}} \|y_i\|^2 \right). \tag{1.30}$$

Modele SVD++ są tu opisane jedynie jako ciekawe rozszerzenie zwykłych modeli SVD i nie będą zbadane empirycznie.

Dalsze usprawnienia

Modele, które były używane w systemach rekomendacyjnych biorących udział w konkursie Netflix'a zawierały zdecydowanie więcej parametrów niż te opisane dotychczas w niniejszej pracy. Bardzo ciekawe było uwzględnienie w modelu daty wystawienia danego rankingu – pozwalało ono na oddzielenie sygnału związanego z tym, że w danym momencie jakiś film był bardzo popularny i jego oceny były w tym okresie zawyżane. Niestety objętość tej pracy nie pozwala na przytoczenie pełnego rozumowania, które doprowadziło do uwzględnienia czasu w modelu. Zainteresowany tym czytelnik może sięgnąć po np. [28] lub [18].

Kolejnym ciekawym rozszerzeniem było uwzględnienie liczby rankingów wystawionych przez użytkownika danego dnia. Wpływ tego czynnika można tłumaczyć w następujący sposób: jeśli użytkownik ocenił danego dnia dużo filmów, to część z nich musiał

widzieć na długo przed wystawieniem oceny, co powoduje, że mógł niedokładnie pamiętać jak bardzo podobał mu się dany film. Autorzy modeli stwierdzili, że ten czynnik bardziej wpływa na zmienność w ocenach danego filmu, a nie ocenach wystawiany przez danego użytkownika. Ostatecznie do modelu dodano parametr $b_{i,f_{ui}}$, gdzie f_{ui} oznacza liczbę rankingów (dokładnie to podłogę z jej logarytmu) wystawionych przez użytkownika u w dniu wystawienia rankingu r_{ui}. Ponownie po dalsze szczegóły odsyłamy do [28] lub [18].

Rozszerzenia opisane w tym rozdziale nie będą zbadane empirycznie.

Rozdział 2

Metody obliczeniowe

2.1. Obliczanie podobieństw grafowych

W rozdziale 1.3.2 zaprezentowano metodę liczenia podobieństwa przedmiotów lub użytkowników, w której tworzy się z nich graf, a podobieństwo dwóch obiektów wyznacza na podstawie ścieżek różnej długości pomiędzy dwoma obiektami. Przypomnijmy, że w tym grafie wierzchołkami są przedmioty i użytkownicy natomiast krawędź łączy dwa wierzchołki tylko wtedy, jeśli jeden reprezentuje użytkownika, który nabył (ocenił) przedmiot reprezentowany przez drugi wierzchołek. Podobieństwo dwóch przedmiotów (lub studentów) x i y oblicza się ze wzoru

$$GraphSim(x,y) = \sum_{i=1}^{n} \alpha^i \cdot paths(x,y,i), \tag{2.1}$$

gdzie $\alpha \in (0,1)$, n to maksymalna długość ścieżek, które bierzemy pod uwagę, a $paths(x,y,i)$ oznacza liczbę ścieżek pomiędzy wierzchołkami reprezentującymi x oraz y zawierających dokładnie i wierzchołków.

Powstaje pytanie, jak w sensowny sposób obliczyć sumę w równaniu 2.1. Okazuje się, że jest na to dosyć prosty sposób, za pomocą którego oblicza się podobieństwa dla wszystkich par obiektów na raz.

Zacznijmy od utworzenia macierzy sąsiedztwa dla tego grafu, przy czym wierzchołki sortujemy tak, żeby pierwsze N wierzchołków reprezentowało przedmioty, a następne M wierzchołków użytkowników. Powstaje macierz blokowa o następującej postaci:

$$S = \left(\begin{array}{c|c} P & Z_u \\ \hline Z_l & U \end{array}\right) = \left(\begin{array}{c|c} 0 & Z_u \\ \hline Z_l & 0 \end{array}\right),$$

gdzie P to macierz sąsiedztwa między przedmiotami rozmiaru $N \times N$, U to macierz sąsiedztwa między użytkownikami rozmiaru $M \times M$, a macierze Z_u i Z_l to macierze sąsiedztwa pomiędzy użytkownikami i przedmiotami rozmiarów $N \times M$ i $M \times N$ odpowiednio (Z_u=Z_l^T). Ze sposobu konstrukcji grafu wynika, że macierze U i P są macierzami zerowymi. Kolejną obserwacją wynikającą ze sposobu konstrukcji grafu (widać to też

z macierzy sąsiedztwa) jest to, że powstał graf dwudzielny, a zatem w sumie z równania 2.1 wyrażenia $paths(x, y, i)$ z nieparzystą wartością i będą równe zero (przez długość ścieżki rozumiemy liczbę zawartych w niej krawędzi). Przy takiej reprezentacji danych macierz sąsiedztwa podniesiona do i-tej potęgi zawiera liczbę ścieżek długości i pomiędzy wierzchołkami, zatem by otrzymać potrzebne informacje należy policzyć sumę:

$$\sum_{i=1}^{n} \alpha^i S^i. \tag{2.2}$$

Mimo, że rozwiązanie wydaje się bardzo proste, to nie do końca takie jest, ponieważ mnożenie (a więc też potęgowanie) macierzy jest trudnym problemem. Dość obiecującym pomysłem jest przekształcenie sumy 2.2 do postaci:

$$\begin{aligned}
&\sum_{i=1}^{n} \alpha^i S^i = \\
&= (I - \alpha S)^{-1}(I - \alpha S) \sum_{i=1}^{n} \alpha^i S^i = \\
&= (I - \alpha S)^{-1} \left(\sum_{i=1}^{n} \alpha^i S^i - \sum_{i=2}^{n+1} \alpha^i S^i \right) = \\
&= (I - \alpha S)^{-1}(\alpha S - \alpha^{n+1} S^{n+1}).
\end{aligned} \tag{2.3}$$

Mając taką postać wystarczy znać $n + 1$ potęgę macierzy S oraz umieć odwrócić macierz $I - \alpha S$. To pierwsze zadanie można wykonać używając potęgowania binarnego, co zmniejsza ilość kosztownych mnożeń. Okazuje się jednak, że takie rozwiązanie niekoniecznie musi być w tym wypadku szybsze od policzenia tej sumy wyraz po wyrazie. W języku R można skorzystać z pakietów służących do operowania na rzadkich macierzach. Macierz S podniesiona do 4 lub 5 potęgi zazwyczaj nie zawiera już zer w dwóch z czterech bloków (pozostałe dwa są zerowe). Wyliczając kolejne potęgi macierzy mnożymy macierz bardzo rzadką przez macierz gęstą, co jest robione stosunkowo szybko. W przypadku gdybyśmy chcieli potęgować binarnie musielibyśmy dokonać mnożeń bloków nie zawierających zer, co okazuje się dużo wolniejsze, nawet jeśli uwzględni się to, że w drugim przypadku mnożeń należy wykonać więcej. Metoda z potęgowaniem binarnym mogłaby być szybsza, gdybyśmy chcieli podnieść macierz do jakiejś naprawdę dużej potęgi. Dodatkowym zyskiem jest to, że nie musimy liczyć odwrotności macierzy $I - \alpha S$.

Wiedząc, że interesują nas tylko parzyste potęgi macierzy, potęgę i-tą postaci

$$S^i = \left(\begin{array}{c|c} P_i & 0 \\ \hline 0 & U_i \end{array} \right)$$

możemy obliczyć, znając wzory na bloki P_i i U_i, które są następujące:

$$P_i = P_{i-2} Z_u Z_l, \tag{2.4}$$

$$U_i = U_{i-2} Z_l Z_u, \tag{2.5}$$

przy czym $U_0 = P_0 = I$.

2.2. Metody minimalizacji funkcji

W rozdziale 1.3.5 proces uczenia modelu polegał na minimalizacji pewnej funkcji wielu zmiennych. Mimo iż funkcje w równaniach 1.24 lub 1.28 nie są jakoś strasznie skomplikowane, to jednak olbrzymia liczba parametrów sprawia, że nie da się znaleźć rozwiązania analitycznie. W przypadku funkcji z równania 1.24 można w sposób przybliżony znaleźć minimum rozdzielając obliczanie b_u od b_i kładąc kolejno

$$b_i = \frac{\sum_{u \in \mathcal{U}_i}(r_{ui} - \mu)}{\lambda_4 + |\mathcal{U}_i|} \tag{2.6}$$

oraz

$$b_u = \frac{\sum_{i \in \mathcal{I}_u}(r_{ui} - \mu - b_i)}{\lambda_5 + |\mathcal{I}_u|}, \tag{2.7}$$

gdzie λ_4 oraz λ_5 są pewnymi metaparametrami. Oczywiście tak otrzymane wyniki ciągle mogą być daleko od minimum, ale mogą stanowić dobry punkt startowy dla innych metod optymalizacyjnych.

W dalszej części tego rozdziału przedstawiono stochastyczną metodę najszybszego spadku, która była używana podczas dopasowywania modeli opisanych w rozdziale 3.4. Pierwsza sekcja zawiera wprowadzenie do metody najszybszego spadku, natomiast druga stanowi już właściwy opis stochastycznej metody najszybszego spadku.

Warto jeszcze dodać, że dzięki parametrom regularyzacyjnym, wraz ze wzrostem normy wektora argumentów do nieskończoności wartość minimalizowanych funkcji (wskazanych w rozdziale 1.3.5) również rośnie do nieskończoności. Ponadto funkcje są ciągłe, zatem mają minima lokalne i jedno z tych minimów jest globalne. Metody przedstawione w dalszej części tego rozdziału gwarantują znalezienie jakiegoś minimum, natomiast nie gwarantują, że będzie ono globalne. W celu wybrania najlepszego z minimów można opisane algorytmy wykonać kilka razy z różnymi punktami startowymi.

2.2.1. Metoda najszybszego spadku

Metoda najszybszego spadku należy do rodziny optymalizacyjnych metod spadkowych. Załóżmy, że chcemy zminimalizować funkcję $f : \mathbb{R}^n \to \mathbb{R}$ klasy $\mathcal{C}^1$ na $\mathbb{R}^n$. Startując od pewnego punktu x_0 generujemy kolejne $x_1, x_2, x_3, ...$, tak żeby $f(x_k) > f(x_{k+1})$. Metody spadkowe to takie, w których kolejny punkt x_{k+1} jest zadany wzorem:

$$x_{k+1} = x_k + \alpha_k d_k, \tag{2.8}$$

gdzie $\alpha_k > 0$ oraz d_k jest kierunkiem spadku, tzn.

$$\begin{aligned} Df(x_k)d_k < 0 \text{ jeśli } Df(x_k) \neq 0, \\ d_k = 0, \text{ jeśli } Df(x_k) = 0, \end{aligned} \tag{2.9}$$

gdzie $Df(x_k) = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \cdots, \frac{\partial f}{\partial x_n}\right)$ – gradient funkcji. Metody najszybszego spadku to takie, w których d_k jest równoległe do $Df(x_k)$ (zazwyczaj $d_k = -\frac{Df(x_k)}{\|Df(x_k)\|}$).

Pozostaje jeszcze dobór α_k. Trzy najczęściej stosowane reguły to:

- reguła minimalizacji: wybierz α_k takie że $f(x_k + \alpha_k d_k) = \min_{\alpha \geqslant 0} f(x_k + \alpha d_k)$.
- reguła ograniczonej minimalizacji: ustalmy $A > 0$. Wybierz α_k takie że $f(x_k + \alpha_k d_k) = \min_{\alpha \in [0,A]} f(x_k + \alpha d_k)$.
- reguła Armijo: ustalmy $s > 0$ i $\beta, \sigma \in (0, 1)$. Połóżmy $\alpha_k = \beta^{m_k} s$, gdzie m_k jest najmniejszą liczbą całkowitą nieujemną m spełniającą nierówność $f(x_k) - f(x_k + \beta^m s\, d_k) \geqslant -\sigma \beta^m s\, Df(x_k) d_k$.

2.2.2. Stochastyczna metoda najszybszego spadku

Stochastyczna metoda najszybszego spadku jest odmianą metody najszybszego spadku opisanej w rozdziale 2.2.1. Jej stosowanie w systemach rekomendacyjnych zostało spopularyzowane przez Simona Funka podczas konkursu Netflix'a (patrz [8]). Stochastyczna metoda najszybszego spadku jest używana tam, gdzie minimalizowaną funkcję można zapisać w postaci sumy różniczkowalnych funkcji (np. zadanie najmniejszych kwadratów, estymacja największej wiarogodności):

$$f(x) = \sum_{i=1}^{m} f_i(x), \tag{2.10}$$

gdzie najczęściej każdy składnik sumy jest związany z jedną obserwacją. Orginalna metoda najszybszego spadku wymagałaby obliczenia gradientu powyższej sumy funkcji, co w przypadku kiedy m jest duże może być bardzo skomplikowane i czasochłonne. W stochastycznej metodzie najszybszego spadku w pojedynczym kroku gradient funkcji f jest przybliżany przez gradient jednej z sumowanych funkcji:

$$x_{k+1} = x_k - \alpha D f_i(x_k). \tag{2.11}$$

Zbiór treningowy jest przebiegany w losowej kolejności i w każdym kroku oblicza się gradient funkcji związanej z aktualnie przetwarzaną obserwacją. Warto jeszcze zauważyć, że w stochastycznej metodzie najszybszego spadku najczęściej parametr odpowiadający za szybkość uczenia (α) jest ustalany przed każdą iteracją przez zbiór treningowy i nie zmienia się w jej trakcie.

W celu ułatwienia zrozumienia algorytmu zapiszemy w pseudokodzie jego wersję, która szuka minimum funkcji z równania 1.28:

- wybierz punkt startowy b_*, q_*, p_*
- wybierz szybkość uczenia α
- dopóki jest to pierwsza iteracja lub wartość funkcji zmniejszyła się w poprzedniej iteracji powtarzaj:
  - wylosuje kolejność przetwarzania obserwacji
  - dla każdej obserwacji r_{ui} ze zbioru treningowego powtarzaj:

* $e_{ui} = r_{ui} - \hat{r}_{ui}$
* $b_u \leftarrow b_u + \alpha(e_{ui} - \lambda_2 b_u)$
* $b_i \leftarrow b_i + \alpha(e_{ui} - \lambda_2 b_i)$
* $q_i \leftarrow q_i + \alpha(e_{ui} p_u - \lambda_2 q_i)$
* $p_u \leftarrow p_u + \alpha(e_{ui} q_i - \lambda_2 q_u)$
* jeśli zmiana parametrów zwiększyła wartość funkcji wróć do poprzednich wartości parametrów

– jeśli zachodzi taka potrzeba zaktualizuj szybkość uczenia α

Jak widać implementacja jest dosyć prosta i właśnie to jest jedna z głównych zalet stochastycznej metody najszybszego spadku. Ponadto jest ona dosyć efektywna, a w przypadku olbrzymich danych nie musimy mieć ich wszystkich naraz w pamięci.

Rozdział 3

System rekomendacyjny na podstawie danych USOS

Poniższy rozdział zawiera opis zastosowania algorytmów opisanych w sekcji 1.3 do danych pochodzących z systemu USOS. Głównym narzędziem wykorzystanym do analizy był język R. Podczas przygotowywania danych pomocne było także kilka skryptów bash'owych. Wszystkie skrypty używane podczas analiz znajdują się na zamieszczonej do pracy płycie DVD.

W tym rozdziale odstąpimy od konwencji przyjętej w sekcji 1.3.1 i będziemy używać określeń student, kurs, ocena zamiast, użytkownik, przedmiot, ranking.

Pierwsze dwie części tego rozdziału zawierają opis danych i procesu ich przygotowywania, do tego by były użyteczne do dalszych analiz. Kolejna sekcja opisuje sposób w jaki weryfikowano skuteczność prognoz poszczególnych metod. Ostatnie dwie części zawierają opis modelowania przeprowadzonego na danych USOS i wyniki analiz.

3.1. Opis danych

Zbiór danych zawiera informację o ocenach studentów, którzy w latach 1999-2009 studiowali na wydziale Matematyki, Informatyki i Mechaniki Uniwersytetu Warszawskiego. Każda obserwacja poza oceną zawiera informacje o studencie takie jak: identyfikator studenta w bazie USOS, jego płeć i datę urodzenia. Z powyższych informacji wykorzystano tylko identyfikator. Oprócz danych studenta każda obserwacja zawiera też informacje o instancji kursu: identyfikator kursu w bazie USOS, nazwę kursu, cykl dydaktyczny oraz dane protokołu z ocenami. Z powyższych jedynie identyfikator okazał się istotny. W zasadzie żadne dane nie mogły być użyte bez wcześniejszej obróbki – należało m.in. zmienić kody studentów i kursów. Pełny opis przygotowania danych znajduje się w sekcji 3.2. Wskaźniki podane w dalszej części tej sekcji będą dotyczyły danych po wstępnej obróbce.

W analizowanych danych znajduje się 186901 obserwacji przy czym 91 z nich to zwolnienia lekarskie z wf-u, które są usuwane z danych. Do analizy pozostaje zatem 186820 obserwacji, na które składają się oceny 4345 studentów i 5008 kursów. Z tego

Wartość szczególna	ZAL	ZWOL-LEK	NZAL	NK
Wartość liczbowa	5	usunięto z danych	2	1

Tablica 3.1: Zamiana szczególnych wartości ocen, na wartości liczbowe. W pierwszym wierszu są szczególne wartości ocen, które mogą znaleźć się w danych, a w drugim wartości liczbowe, na które zostały zamienione.

wynika, że średnio każdy student ma około 43 ocen, a na każdy kurs przypada około 37 obserwacji. Odchylenia od powyższych średnich wartości są jednak bardzo duże – liczba ocen dla jednego studenta waha się od 1 do 169, a dla jednego kursu od 1 do 2448 Dodatkowo, są studenci, którzy z jednego kursu mają kilka ocen (egzaminy poprawkowe, powtarzanie kursu, wf). Unikalnych par student-kurs jest 145920. Daje to średnio 34 oceny na studenta (minimalnie 1, a maksymalnie 114 ocen) i 30 na kurs (minimalnie 1, a maksymalnie 1970 ocen).

3.2. Jak przygotowano i wykorzystano dane

Dane początkowo znajdowały się w wielu plikach – po jednym pliku na kurs. Część z plików zawierała tylko nagłówek (nazwy kolumn), dlatego pierwszym krokiem było usunięcie plików, które zawierały tylko jedną linię. Zadanie to zostały wykonane przez krótki skrypty w bashu, który można znaleźć na dołączonej płycie DVD.

Kolejnym problemem był fakt, że w ciągu lat 1999-2009 niektóre przedmioty były prowadzone w dwóch równoległych potokach, a kody części zmieniły się. W związku z tym oceny z jednego kursu można było znaleźć w wielu plikach i miały one przypisane różne identyfikatory kursu. Niestety nie ma prostego schematu zmian nazw kursów i należało ręcznie wyszukać kursy, które były rozbite na kilka innych i dopiero wtedy automatycznie je połączyć.

Po wykonaniu powyższych czynności można już było przejść do obróbki danych przeprowadzonej przez skrypt R-owy. Pierwszą czynnością było wczytanie wszystkich obserwacji do jednej ramki danych, a następnie zamiana id studentów i kursów, tak żeby założenia z sekcji 1.3.1 były spełnione.

Kolejnym krokiem był podział danych na zbiór treningowy i testowy (patrz sekcja 3.3), po którym dla każdej pary student-cykl dydaktyczny została policzona liczba kursów, na które student uczęszczał w danym semestrze.

Kolejnym problemem był fakt, że część ocen nie miała wartości liczbowej, tylko specjalny kod (np. "NZAL"). Opis znaczenia danych kodów i wartości na które je zamieniono znajduje się w tablicy 3.1.

W przypadku, gdy student miał z danego kursu więcej niż jedną ocenę policzono średnią ocenę z danego przedmiotu, przy czym zbiór treningowy i testowy potraktowano oddzielnie.

Na koniec wybrano te kolumny, które były istotne dla dalszych analiz i zapisano dane do trzech plików. Opis formatu plików znajduje się w tablicy 3.2. Taki też format jest

wymagany przez skrypty dokonujące dalszych analiz. Kolumny powinny być oddzielone separatorem ";".

3.3. Sposoby weryfikacji

W celu porównania różnych algorytmów przewidujących oceny należało sprawdzić jak bardzo prognozy odbiegają od rzeczywistych ocen. Oczywiście nie można było użyć do tego celu zbioru na którym algorytmy się uczyły i dlatego przed rozpoczęciem analiz wydzielono część danych jako zbiór testowy. Opis sposobu w jaki tego dokonano znajduje się w sekcji 3.3.1. Kolejne sekcje tego rozdziału zawierają opis miar, które zastosowano do oceny tego, jak bardzo prognozy odbiegają od rzeczywistości oraz opis innych sposobów oceny tego, czy dany metoda daje dobre wyniki.

3.3.1. Zbiór testowy

Głównym celem analiz przeprowadzanych w tej pracy jest dostarczenie studentom informacji o tym jakich ocen mogą się spodziewać z danego kursu. Mniej ważne jest natomiast przewidzenie jakie oceny będą wystawione na tym kursie. W rzeczywistości to drugie zadanie wymagałoby wiedzy o tym, jacy studenci będą na dany kurs uczęszczać. W związku z tym do zbioru testowego trafiały oceny studenta z ostatniego roku, w którym dany student otrzymał jakieś oceny. To oczywiście sprawia, że część ocen w zbiorze treningowym została wystawiona później niż niektóre oceny ze zbioru testowego, które mamy przecież prognozować. Ten fakt może budzić zastrzeżenia, ponieważ system rekomendacyjny przewidujący jakąś ocenę nie ma dostępu do danych z przyszłości i jego ocena w tak skonstruowanym eksperymencie może być nieadekwatna. Podobne podejście zastosowano jednak w konkursie Netflix'a ([25]; na fakt ten zwrócono uwagę w [1]), a dodatkowo wzięcie do zbioru testowego ocen wystawionych po jakieś określonej dacie sprawiłoby, że dużej części studentów ze zbioru testowego w ogóle nie byłoby w zbiorze treningowym (i vice versa). To zresztą i tak był pewien problem – dla części studentów dostępne były dane z tylko jednego roku, i dlatego nie znaleźli się oni w zbiorze treningowym a co za tym idzie jakakolwiek prognoza na ich temat nie była możliwa. Podobnie rzecz miała się z kursami – część z nich nie pojawiła się w zbiorze treningowym. Postanowiono żeby tych studentów i kursów w ogóle nie brać pod uwagę, co nieznacznie pomniejszyło zbiór testowy.

3.3.2. Miary poprawności prognozy

W celu określenia tego, jak skuteczne są prognozy danego algorytmu najprościej jest obliczyć za pomocą jakiejś miary to, jak bardzo przewidywane oceny różnią się od ocen rzeczywistych. Liczbowe określenie skuteczności algorytmu pozwala też na porównywanie różnych metod. Oczywiście ocena jakości tak naprawdę zależy od doboru miary, a poza tym sprowadzenie jakości do jednej liczby jest dużym uproszczeniem. Jest to jednak dobry punkt wyjściowy do dalszej analizy (patrz sekcja 3.3.3) i dlatego przytaczamy tutaj parę miar, które zostały użyte do oceny proponowanych algorytmów.

Nazwa pliku:	studentMapping
Nazwy kolejnych kolumn	Zawartość kolumny
nowe	identyfikator studenta używany w analizach
stare	orginalny identyfikator studenta
nie.liczyc	czy dla studenta należy liczyć prognozę (możemy tego nie chcieć, bo np. nie ma go w zbiorze treningowym)

Nazwa pliku:	courseMapping
Nazwy kolejnych kolumn	Zawartość kolumny
nowe	identyfikator kursu używany w analizach
stare	orginalny identyfikator kursu
nie.liczyc	czy dla kursu należy liczyć prognozę (możemy tego nie chcieć, bo np. nie ma go w zbiorze treningowym)
nazwa	pełna nazwa kursu

Nazwa pliku:	colSelectionBig
Nazwy kolejnych kolumn	Zawartość kolumny
os_id	identyfikator studenta używany w analizach
prz_kod	identyfikator kursu używany w analizach
ocena	ocena danego studenta z danego kursu
sasiad	zawsze 1 – traktując studentów i przedmioty jako wierzchołki grafu oznacza, że dany student i dany kurs są połączone krawędzią (patrz sekcja 1.3.2)
ile_kurs	na ile kursów uczęszczał student w cyklu dydaktycznym, w którym otrzymał tą ocenę
zb.treningowy	czy ta ocena znajduje się w zbiorze treningowym
zb.testowy	czy ta ocena znajduje się w zbiorze testowym
wartosc.sr	wartość średnia ocen danego studenta z danego kursu

Tablica 3.2: Oczekiwany format plików z danymi. Większość skryptów wymaga trzech plików: *studentMapping* i *courseMapping*, w których znajdują się mapowania orginalnych identyfikatorów studentów i kursów na nowe identyfikatory (kolejne liczby naturalne) oraz *colSelectionBig*, w którym znajdują się oceny studentów z poszczególnych kursów wraz z pewnymi dodatkowymi informacjami na temat tych ocen (np. czy jest ona w zbiorze testowym, czy treningowym).

Jedną z podstawowych miar tego jak bardzo prognoza odbiega od rzeczywistych wyników jest błąd średniokwadratowy (ang. *mean squared error*) lub jego pierwiastek (ang. *root mean squared error*). W konkursie Netflix'a używany był ten drugi i dlatego my też będziemy go używać. Jest on zdefiniowany następująco:

$$RMSE(\hat{r}) = \sqrt{\frac{\sum_{r_{ui} \in R_{test}} (\hat{r}_{ui} - r_{ui})^2}{|R_{test}|}}. \tag{3.1}$$

Drugą dosyć często używaną miarą jakości prognozy jest błąd średni (ang. *mean absolute error*):

$$MAE(\hat{r}) = \frac{\sum_{r_{ui} \in R_{test}} |\hat{r}_{ui} - r_{ui}|}{|R_{test}|}. \tag{3.2}$$

Pierwiastek błędu średniokwadratowego w stosunku do błędu średniego ma tą zaletę, że zaniedbuje małe odchylenia od rzeczywistości, a wyolbrzymia te większe.

Czasem studenta może tylko interesować, to czy z danego przedmiotu dostanie ocenę powyżej swojej dotychczasowej średniej, czy poniżej i w efekcie czy ta średnia zostanie zawyżona, czy zaniżona. Dlatego można też policzyć trafność przewidywania tego, po której stronie średniej studenta znajdzie się ocena z danego przedmiotu (przyjmijmy roboczą nazwę MSPA – *mean side prediction accuracy*):

$$MSPA(\hat{r}) = \frac{|\{r_{ui} \in R_{test} | (r_{ui} > \bar{r}_{ui} \wedge \hat{r}_{ui} > \bar{r}_{ui}) \vee (r_{ui} < \bar{r}_{ui} \wedge \hat{r}_{ui} < \bar{r}_{ui})\}|}{|R_{test}|}. \tag{3.3}$$

Ponieważ system ma rekomendować studentowi przedmioty, z których może on dostać najwyższe oceny, to można zaniedbać to jak bardzo prognozy były podobne do rzeczywistości, a zwrócić uwagę jedynie na uszeregowanie prognoz i rzeczywistych ocen. Jednym ze sposobów na zbadanie, czy między dwoma zmiennymi występuje monotoniczna zależność jest policzenie współczynnik korelacji rang Spearmana. Współczynnik ten przyjmuje wartości z przedziału $[-1, 1]$, przy czym wartość 1 oznacza, że jedna zmienna jest ściśle rosnącą funkcją drugiej, a wartość -1, że funkcja ta jest ściśle malejąca. Jeśli $R(x_i)$ oznacza rangę wartości zmiennej x_i:

$$R(x_i) = \sum_{k=|\{x_j : x_j < x_i\}|+1}^{|\{x_j : x_j \leqslant x_i\}|} \frac{k}{|\{x_j : x_j \leqslant x_i\}| - |\{x_j : x_j < x_i\}|} \tag{3.4}$$

oraz

$$R(x) = (R(x_1), R(x_2), ..., R(x_n)), \tag{3.5}$$

to współczynnik korelacji rang pomiędzy zmiennymi x i y wyraża się następująco:

$$r_S(x, y) = corr(R(x), R(y)). \tag{3.6}$$

W naszym przypadku, tzn. do zbadania, czy sugerowana kolejność przedmiotów jest właściwa, wzoru 3.6 używa się w następujący sposób:

$$SRCC(\hat{r}) = \frac{\sum_{u \in R_{student,test}} r_S(\hat{r}_u, r_u)}{|R_{student,test}|}, \tag{3.7}$$

gdzie $\hat{r}_u$ i r_u oznaczają odpowiednio prognozę i rzeczywiste wartości dla studenta u, a $R_{student,test}$ to zbiór wszystkich studentów, którzy mają jakieś oceny w zbiorze testowym.

Główną miarą użytą w tej pracy do oceny algorytmów był pierwiastek błędu średniokwadratowego, ale pozostałe również zostały obliczone i są prezentowane częściowo w rozdziale 3.5, a w pełni w dodatku A i w plikach z wynikami znajdującymi się na dołączonej do pracy płycie DVD.

3.3.3. Inne sposoby oceny algorytmów

Posługiwanie się miarami opisane w poprzedniej sekcji do oceny algorytmów ma tę wadę, że sprowadzając wynik do jednej liczby pozbawia nas wielu informacji na temat prognoz generowanych przez algorytm. Do takich informacji należą np.:

- w jakich sytuacjach algorytm radzi sobie gorzej, a w jakich lepiej (może się okazać, że jeden algorytm lepiej przewiduje oceny, które w rzeczywistości są niskie, a inny te, które są wysokie);

- dlaczego algorytm został oceniony źle albo dobrze (może się okazać, że jedna bardzo zła prognoza sprawia, nawet jeśli inne są doskonałe, że metoda jest oceniona gorzej niż taka, która większość prognoz ma kiepskich, ale też nie ma żadnych skrajnie złych);

- z czego wynika, że algorytm prognozuje źle lub dobrze.

Aby znaleźć odpowiedź na dwa pierwsze pytania należy dokładniej przyjrzeć się otrzymanym rezultatom. Oczywiście nie jest możliwe zbadanie każdej prognozy osobno, ale można zebrać jakieś zbiorcze informacje i na ich podstawie spróbować ocenić algorytmy. Do takich zbiorczych informacji może należeć np. sprawdzenie jak rozkładają się prognozy w zależności od rzeczywistej oceny albo przyjrzenie się rozkładowi różnic pomiędzy prognozami, a prawdziwymi ocenami. Oczywiście ocena takich informacji zebranych w formę tabelek jest również trudna i dlatego warto posiłkować się różnymi graficznymi metodami przedstawiania takich informacji.

W przypadku ostatniego pytania sprawa jest nieco trudniejsza, ponieważ nie mamy wielu pośrednich wyników, na podstawie których można by to ocenić. Dla metody slope one jest to macierz "dev" (patrz rozdział 1.3.4), dla algorytmów najbliższych sąsiadów macierze podobieństwa (i wag, które jednak zależą od macierzy podobieństwa), natomiast w przypadku modelowania mamy parametry bazowe i macierze cech, dla których ciężko jest ocenić bezpośredni wpływ na jakość prognoz. Dodatkowo ze względu na rozmiary tych pośrednich wyników trudno je interpretować bezpośrednio i dlatego znowu trzeba posłużyć się metodami graficznymi, które oczywiście nie mogą być do końca obiektywne i dawać jednoznacznych odpowiedzi.

3.4. Modelowanie na danych USOS

W rozdziale 1.3.5 opisano modele tworzone dla danych zawierających oceny filmów. Niestety większość tych modeli nie mogła być zastosowana dla danych pochodzących z systemu USOS. Dodatkowym problemem był fakt, że jednym z założeń tej pracy było przeprowadzanie analiz w języku R, który jest mniej wydajny od narzędzi stosowanych przez uczestników konkursu Netflix'a. W efekcie porządne dopasowanie bardziej skomplikowanych modeli zajęłoby bardzo dużo czasu. Ostatecznie dopasowano jedynie kilka podstawowych modeli, których opis znajduje się poniżej.

Pierwsze dopasowywane modele zawierały tylko parametry bazowe. Innymi słowy minimalizowana była funkcja:

$$\sum_{(u,i,r_{ui})\in\mathcal{R}} (r_{ui} - \mu - b_u - b_i)^2 + \lambda_1 \left(\sum_{u\in\mathcal{U}} b_u^2 + \sum_{i\in\mathcal{I}} b_i^2 \right), \tag{3.8}$$

gdzie parametr regularyzacyjny λ_1 przyjmował kolejno wartości 0.001, 0.001, 0.1 oraz 1, a znalezione minima to odpowiednio: 68264.57, 68286.82, 68674.48 i 72217.48. Minimum szukano za pomocą stochastycznej metody najszybszego spadku (opisanej w rozdziale 2.2.2), przy czym parametr odpowiadający za szybkość uczenia był tak dostosowywany by zapewnić jak najszybsze dopasowanie modelu. Jedna iteracja (przejrzenie wszystkich elementów w zbiorze treningowym) trwała około 32 minut.

Drugim typem dopasowywanych modeli były te oznaczone w rozdziale 1.3.5 jako SVD, z minimalizowaną funkcją postaci:

$$\begin{aligned} &\sum_{(u,i,r_{ui})\in\mathcal{R}} (r_{ui} - \mu - b_u - b_i - q_i p_u)^2 + \\ &\lambda_2 \left(\sum_{u\in\mathcal{U}} b_u^2 + \sum_{i\in\mathcal{I}} b_i^2 + \sum_{u\in\mathcal{U}} \|p_u\|^2 + \sum_{i\in\mathcal{I}} \|q_i\|^2 \right), \end{aligned} \tag{3.9}$$

gdzie parametr regularyzacyjny λ_2 ponownie przyjmował wartości 0.001, 0.001, 0.1 oraz 1, natomiast parametr f oznaczający wymiar przestrzeni ukrytych cech przyjmował kolejno wartości 10, 20 i 40. Tabela 3.3 przedstawia osiągnięte minima dla wszystkich kombinacji parametrów. Czas trwania iteracji zależał od wartości parametru f i wynosił: 50 minut dla f równego 10 i 20 oraz 71 minut dla f równego 40.

3.5. Wyniki

W tej sekcji zostaną przedstawione najciekawsze wyniki otrzymane z przeprowadzonych analiz. Kompletne, chociaż suche, wyniki znajdują się na dołączonej do pracy płycie DVD (pod koniec każdej z sekcji 3.5.1, 3.5.2, 3.5.3 znajduje się informacja o tym, w jakim katalogu na płycie można znaleźć kompletne wyniki dla opisywanych metod) oraz w dodatku A.

λ_2 \ f	10	20	40
0.001	40851.00	43086.48	30038.58
0.01	38304.96	35032.97	32216.39
0.1	44147.17	33575.47	37778.02
1	79960.43	88179.21	102607.20

Tablica 3.3: Minima funkcji z równania 3.9 osiągnięte podczas uczenia modeli SVD dla różnych kombinacji parametrów λ_2 i f. Jedyną widoczną regularnością jest to, że w momencie, kiedy parametr λ_2 wynosi 1, to minimum funkcji zdecydowanie się zwiększa. Duży wymiar przestrzeni ukrytych cech (f) zazwyczaj pozwala lepiej dopasować się do danych i zmniejszyć minimum, chociaż w przypadku $\lambda_2 = 0.1$ lepiej dopasowuje się przestrzeń z $f = 20$.

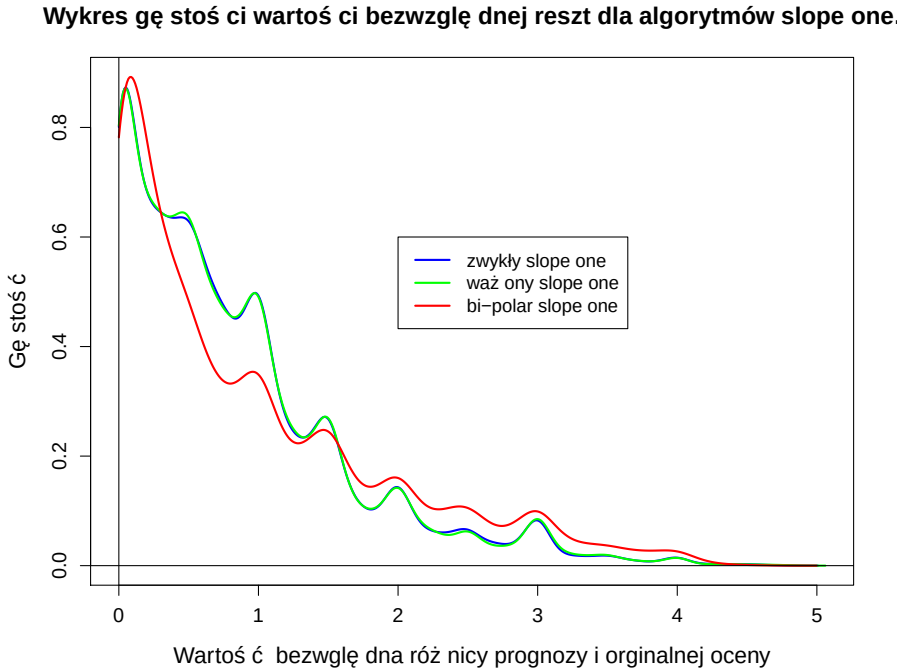
	slope one	ważony slope one	bi-polar slope one
RMSE	**1.15**	**1.15**	1.34
MAE	**0.91**	**0.91**	0.97
MSPA	**0.68**	**0.68**	0.58
SRCC	**0.41**	**0.41**	0.36

Tablica 3.4: Wyniki dla algorytmu slope one. Na dwóch miejscach po przecinku nie widać różnicy pomiędzy odmianami zwykłą i ważoną, natomiast w świetle powyższych kryteriów odmiana bi-polar wypada gorzej – jej błędy średnie są większe (dwa pierwsze wiersze), rzadziej jest w stanie odgadnąć, czy ocena będzie mniejsza, czy większa od średniej studenta (wiersz 3), a proponowana przez nią kolejność przedmiotów bardziej odbiega od tej rzeczywistej (ostatni wiersz).

3.5.1. Slope one

Analizując wyniki algorytmów slope one w oczy rzuca się przede wszystkim fakt, że odmiany zwykła i ważona zazwyczaj były w stanie zaproponować jakąś ocenę – jedynie w 74 przypadkach (0.4% rozmiaru zbioru testowego) algorytmy te były bezradne. Jest to moim zdaniem wielka zaleta tych algorytmów, ponieważ metoda, która jest w stanie wygenerować prognozy tylko dla niewielkiej części zapytań, nawet jeśli są one idealne, jest bezużyteczna. Algorytmy najbliższych sąsiadów (poza tymi opartymi o podobieństwo grafowe) zazwyczaj miały problemy z generowaniem prognoz w większej ilości przypadków (patrz sekcja 3.5.2). Już zresztą wersja bi-polar metody slope one okazała się bezradna w 428 przypadkach (2% całego zbioru testowego).

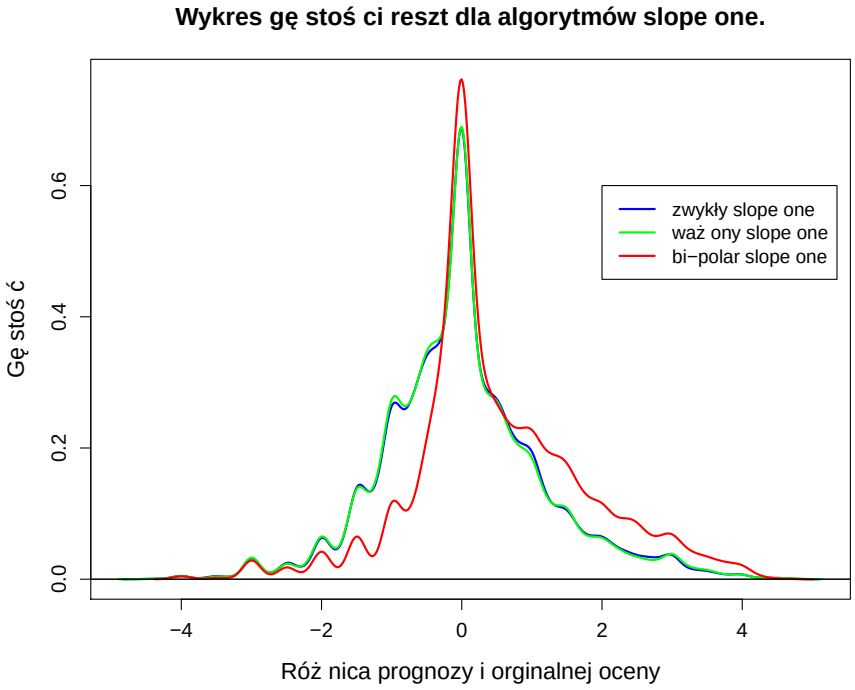
Oczywiście sam fakt, że algorytm potrafi prognozować w prawie każdej sytuacji to za mało, żeby uznać go za dobry. Istotna jest też jakość jego przewidywań. Miary opisane w sekcji 3.3.2 dla algorytmów z rodziny slope one są przedstawione w tabeli 3.4. Biorąc pod uwagę jedynie RMSE i porównując z wynikami metody najbliższych sąsiadów można powiedzieć, że algorytmy zwykły i ważony slope one radzą sobie całkiem dobrze. Dosyć ciekawy jest fakt, że metoda ważona osiąga wyniki bardzo zbliżone do metody zwykłej.



Rysunek 3.1: Gęstość wartości bezwzględnej różnic pomiędzy prognozą, a orginalną oceną dla algorytmów slope one. Rozkład reszt dla metody zwykłej i ważonej jest niemal identyczny. W stosunku do nich, odmiana bi-polar częściej generuje generuje bardzo dobre i bardzo złe prognozy.

Według [20] ta pierwsza powinna zachowywać się lepiej, tymczasem u nas zachowuje się nieznacznie gorzej. Prawdopodobnie wynika to z faktu, że tak naprawdę większość wag jest równa 1, a blisko 80% z nich jest mniejsza od 3. Różnica popularności oparta o dużą liczbę studentów (tzn. z dużą wagą) występuje najczęściej dla przedmiotów obowiązkowych, natomiast my, ponieważ prognozujemy dla kursów z ostatniego roku danego studenta zazwyczaj bierzemy pod uwagę kursy obieralne. To oznacza, że dla większości prognoz w ważonym slope one niewiele się zmienia w stosunku do tego zwykłego. Co więcej, jeżeli np. student nie radził sobie na studiach i był zmuszony zrezygnować z nich wcześniej, to dla niego faktycznie prognozujemy przedmioty obowiązkowe, ale duże wagi mogą zmieniać prognozy na gorsze w skutek tego, że oceny tego studenta nie były związane z jego faktycznymi zdolnościami.

W pracy [20] wspomniano, że ponieważ w wersji bi-polar algorytmu slope one pod uwagę bierzemy tak naprawdę mniejszą ilość par kursów, to nieintuicyjne wydaje się to, że może ona dawać lepsze rezultaty. W doświadczeniach opisanych we wspomnianej pracy to jednak właśnie ta odmiana generowała najdokładniejsze prognozy. Dla danych dotyczących ocen studentów już tak nie było – biorąc pod uwagę jedynie RMSE należy uznać, że metoda ta dawała zauważalnie gorsze wyniki od dwóch pozostałych. Ponadto,



Rysunek 3.2: Gęstość różnic pomiędzy prognozą, a orginalną oceną dla algorytmów slope one. Poza obserwacjami z rysunku 3.1 widać tutaj, że odmiany zwykła i ważona mają tendencję do niedoszacowywania, a wersja bi-polar do przeszacowywania ocen.

jak już wcześniej wspomniano w większej ilości przypadków nie była w stanie wygenerować jakiejkolwiek prognozy. Rysunek 3.1 sugeruje jednak, że ta wersja metody stworzyła bardzo wiele dobrych prognoz, ale też dużo bardzo złych i dlatego według wskaźników z tabeli 3.4 wypada gorzej. Analiza rysunku 3.2 pokazuje, że jeśli odmiana bi-polar się myliła, to miała tendencję do zawyżania oceny. Dokładniejsze zbadanie sprawy wskazuje, że kiepskie prognozy najczęściej dotyczą bardzo niskich ocen (sytuacji, w których student nie zdawał lub nie był klasyfikowany w pierwszym terminie lub w ogóle). To może po części tłumaczyć dlaczego ta odmiana oprócz bardzo dobrych prognoz generuje też wiele bardzo słabych.

Katalog z wynikami

Pełne wyniki opisywane w poprzedniej sekcji można znaleźć na płycie DVD w katalogu "/wyniki/slope one". W kolejnych podkatalogach znajdują się:

- macierze z "różnicami popularności" (patrz sekcja 1.3.4) w katalogu "macierze dev";
- wyniki pomiarów jakości prognoz (RMSE, MAE, MSPA, SRCC) w katalogu "po-

miary";

- prognozy dla elementów ze zbioru testowego w katalogu "prognozy".

3.5.2. Najbliżsi sąsiedzi

W przypadku algorytmów slope one nie mieliśmy do wyboru zbyt wielu możliwości dostrajania metody i w efekcie otrzymaliśmy jedynie trzy odmiany algorytmu. W przypadku algorytmów najbliższych sąsiadów tych możliwości jest dużo więcej, z których w niniejszej pracy zbadano następujące:

- jaką metodę obliczania podobieństwa/odległości wybrać (patrz sekcja 1.3.2);
- czy podobieństwo/odległość obliczać dla studentów, czy kursów;
- czy normalizować rankingi przed obliczeniem podobieństwa/odległości (patrz sekcja 1.3.2);
- czy skurczyć wagi biorąc pod uwagę liczbę obiektów na podstawie, których obliczono podobieństwo/odległość (patrz sekcja 1.3.2);
- czy najbliższych sąsiadów wybierać według metody "k najbliższych", czy powyżej wartości progowej (patrz sekcja 1.3.3);
- jak dobrać ilość najbliższych sąsiadów lub wartość progową;
- czy normalizować rankingi przed obliczeniem prognozy (patrz sekcja 1.3.2).

Dla pewnego wybranego zbioru wartości progowych i liczby najbliższych sąsiadów starano się sprawdzić wszystkie możliwe kombinacje pozostałych parametrów metody. Pominięto oczywiście zestawy, które nie miały sensu (np. korelacja Pearsona zawiera już w sobie normalizację, a miara dostosowany kosinus nie ma sensu w przypadku studentów). Ponadto wpływ kurczenia wag sprawdzono tylko dla kilku wybranych kombinacji pozostałych parametrów.

Zdolność generowania prognoz

Na zdolność generowania prognoz najbardziej wpływa parametr decydujący o tym, kogo zaliczyć do zbioru najbliższych sąsiadów. W przypadku wyboru za pomocą wartości progowej dla poziomów progu większych od 0.2 (analizowano miary podobieństwa o zakresach wartości $[-1, 1]$) dla każdej metody w ponad 10% przypadków nie udało się stworzyć prognozy. Biorąc pod uwagę dobór k najbliższych sąsiadów wyniki są dużo lepsze – w najgorszym przypadku nie udało się stworzyć prognozy dla 5% zapytań, natomiast były algorytmy, które zawsze potrafiły wygenerować jakąś prognozę. Średnio w przypadku doboru k najbliższych przewidywanie nie udawało się w 0.8% przypadków (mediana 0.4%), natomiast w przypadku doboru na podstawie wartości progowej w 27% przypadków (mediana 14%). W dalszych rozważaniach będziemy brali pod uwagę tylko przypadki, kiedy niezdolność do wykonania prognozy była mniejsza od 10%.

nazwa miary	liczone podobieństwa	normalizacja	procent ujemnych wag	procent dodatnich wag
kosinus	studentów	brak	0%	68.6%
kosinus	studentów	centrowanie	19.1%	48.4%
kosinus	studentów	z-score	19.7%	48.8%
kosinus	kursów	brak	0%	4.6%
kosinus	kursów	centrowanie	1.1%	1.9%
kosinus	kursów	z-score	1.7%	2.8%
korelacja Pearsona	studentów	brak	19.1%	48.4%
korelacja Pearsona	kursów	brak	1.1%	1.9%
dostosowany kosinus	kursów	brak	2.3%	2.3%

Tablica 3.5: Procent wag dodatnich i ujemnych dla wybranych metod liczenia podobieństwa. Pierwsze trzy kolumny określają to, z jakim podobieństwem mamy do czynienia (jaką było policzone miarą, czy pomiędzy kursami, czy studentami i czy znormalizowano rankingi). Kolejne mówią o tym jaką część obliczonych podobieństw (a zatem także wag w algorytmach NN) stanowiły podobieństwa dodatnie, a jaką ujemne. Wartości nie sumują się do 100%, ponieważ części podobieństw nie dało się w ogóle obliczyć. Metody z wyższym odsetkiem wag dodatnich zazwyczaj zachowują się lepiej.

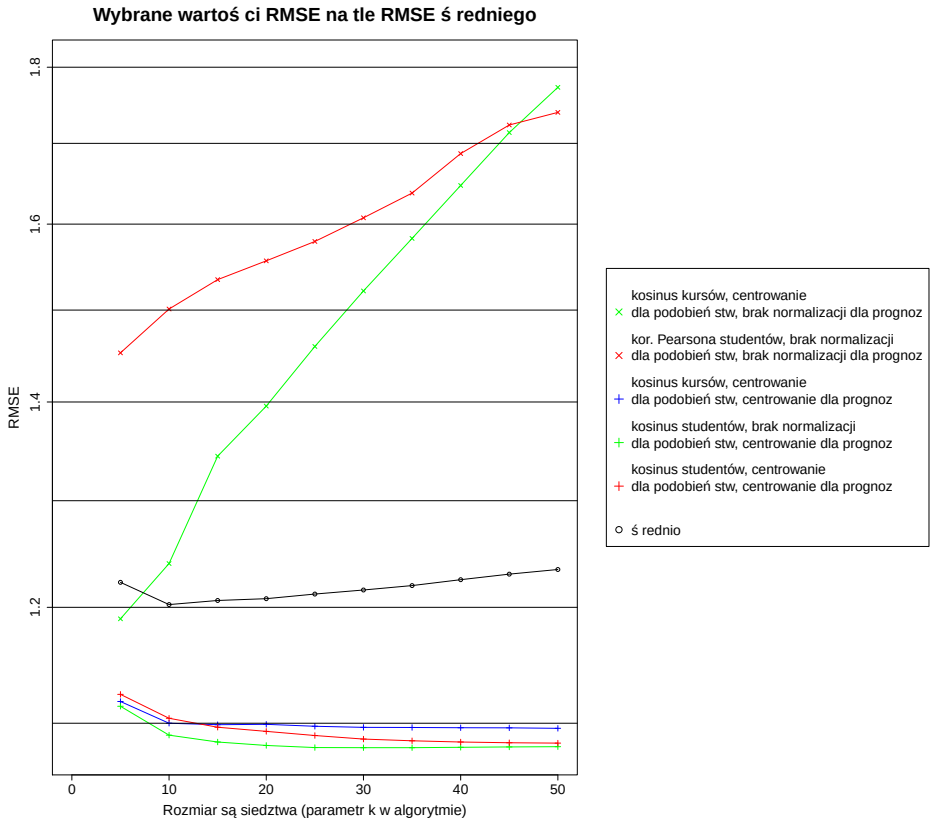
Wyniki

Gdyby nie zastrzeżenie zawarte w poprzednim rozdziale za świetne należałoby uznać te wersje algorytmu, które do zbioru najbliższych sąsiadów zaliczają wszystkich powyżej pewnego dodatniego progu. Ponieważ jednak chcemy, żeby w co najmniej 90% przypadków generował prognozę, to próg najczęściej musi być stosunkowo niski, a przy takich wspomniane metody prognozują już gorzej. Na uwagę zasługuje jedynie wersja oparta o podobieństwo dostosowanego kosinusa pomiędzy kursami – RMSE wynosi 1.1012, co jest gorszym wynikiem od kilkunastu metod opartych o wybór k najbliższych sąsiadów, ale biorąc pod uwagę wszystkie sprawdzone metody z rodziny najbliższych sąsiadów (patrz rysunek 3.3) jest to niezły wynik.

Powyższe porównanie wskazuje na głębszy problem – co zrobić z ujemnymi wagami (takie mogą wystąpić w momencie, gdy zakres wartości miary podobieństwa obejmuje też liczby ujemne). W niniejszej pracy postąpiono zgodnie z sugestią zawartą w [28], tzn. podczas mnożenia rankingu waga była ujemna, ale już w mianowniku podczas normalizacji brano jej wartość bezwzględną. Wyniki metod z progową metodą wyboru sąsiadów wskazują, że ujemne wagi psują jakość rekomendacji, co oczywiście nie dziwi.

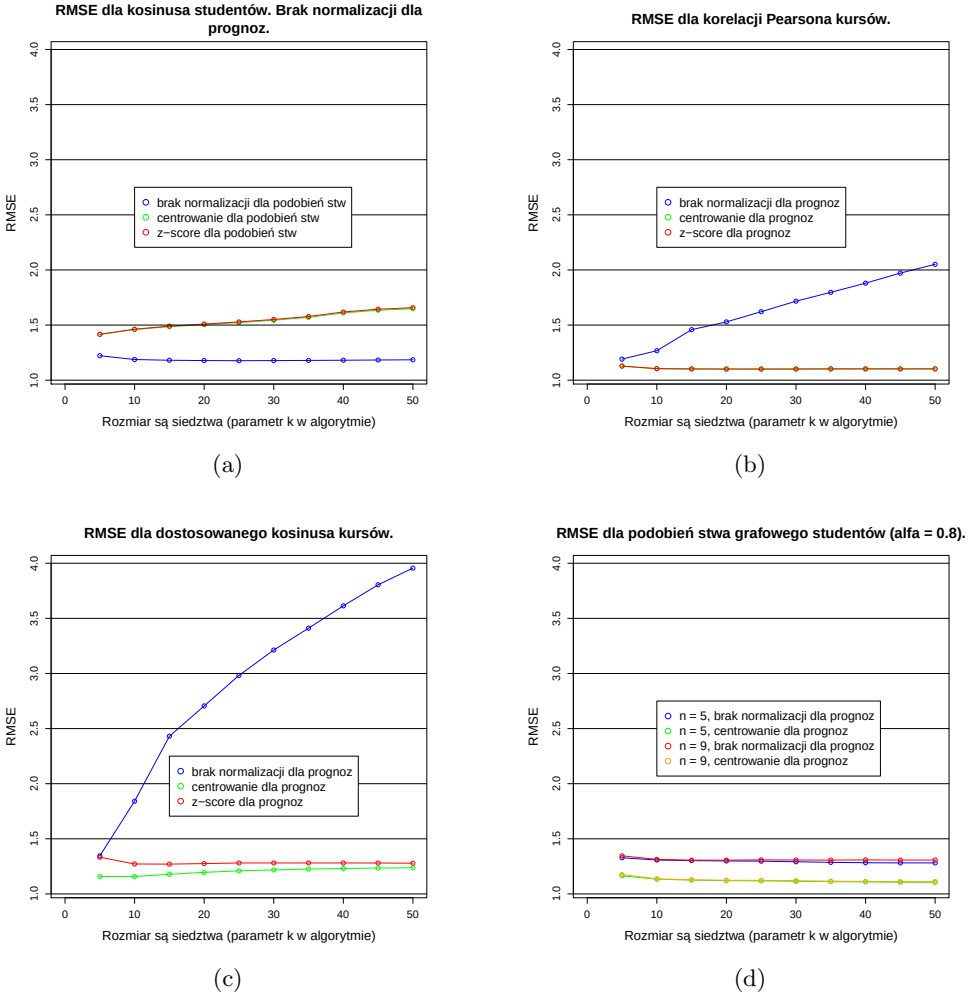
W tym miejscu warto przytoczyć pewne informację dotyczące tego na ilu wspólnych kursach lub studentach opierano podobieństwa, a także ile było wag dodatnich, a ile ujemnych przy generowaniu prognozy. W przypadku podobieństwa kursów jedynie co 25 para miała studentów, którzy uczęszczali na oba kursy (średnio 0.37 wspólnych studentów na parę – było kilka par z dużą liczbą wspólnych studentów), natomiast dla podobieństwa studentów było już dużo lepiej – 67% par miało jakiś wspólny kurs, co

dało średnią w wysokości 6.18 kursów na parę. W tabeli 3.5 znajduje się informacja na temat tego ile procent wag było ujemnych, a ile dodatnich dla danej metody, przy czym przedstawiono tylko te miary, dla których była możliwość otrzymania ujemnych wyników (pominięto odległość euklidesową i podobieństwa grafowe). Oczywiście rankingi zawsze były dodatnie dlatego w przypadku kosinusów nie ma ujemnych wag. Warto też zwrócić uwagę, że kosinus z centrowaniem to po prostu korelacja Pearsona. Zastanawiające może się wydawać dlaczego różne są odpowiadające sobie wartości dla centrowania i normalizacji z-score. Wynika to z tego, że jeśli wariancja była równa zero, to zostawialiśmy orginalne rankingi. Okazuje się, że przedstawione powyżej fakty znajdują odbicie w wynikach – lepsze są te prognozy gdzie mamy większy procent wag dodatnich (patrz rysunek 3.4a).



Rysunek 3.3: Najwyższe i najniższe RMSE na tle RMSE średniego w zależności od rozmiaru sąsiedztwa. Skala na osi Y jest logarytmiczna. Dość ciekawy jest fakt, że dla dobrych metod zwiększanie rozmiaru sąsiedztwo polepsza jakość prognoz, natomiast dla kiepskich pogarsza. Średnia z RMSE dla wszystkich algorytmów z rodziny najbliższych sąsiadów też rośnie wraz ze wzrostem k, ale może to być wywołane przez kilka bardzo złych metod (np. dostosowany kosinus, który akurat tutaj nie jest pokazany).

Na osobne potraktowanie zasługują metody, w których obliczanie podobieństwa znacz-



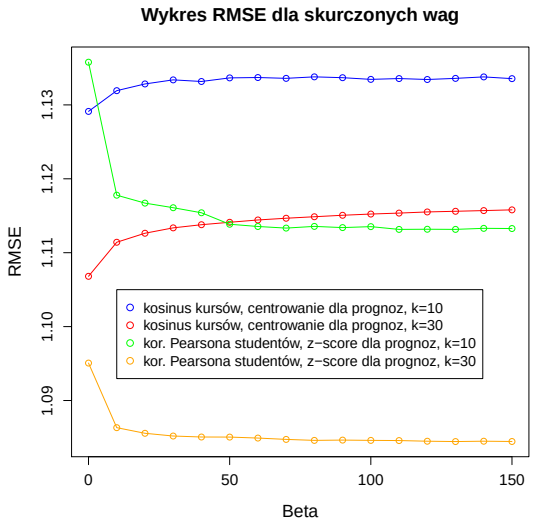
Rysunek 3.4: RMSE dla wybranych metod w zależności od rozmiaru sąsiedztwa. Na kolejnych rysunkach znajdują się różne metody liczenia podobieństwa: (a) – kosinus studentów, (b) – korelacja Pearsona kursów, (c) – dostosowany kosinus kursów, (d) – podobieństwo grafowe studentów dla $\alpha = 8$. Na uwagę zasługują dosyć dobre wyniki podobieństw grafowych z normalizacją dla prognoz oraz fatalne dla dostosowanego kosinusa kursów bez normalizacji dla prognoz.

nie różniło się od wszystkich pozostałych, tzn. te wykorzystujące odległość euklidesową i podobieństwo grafowe. Dla metod grafowych macierze podobieństw zawierały duże wartości, dlatego podczas liczenia prognoz zostały one przeskalowane do przedziału $[0, 1]$. Mogło to nieznaczenie zmienić wyniki przewidywań, bo po podzieleniu przez największą wartość w macierzy pewne wartości mogły być niereprezentowalne (nie zmieniły się oczywiście kolejność sąsiadów i ich wpływ – podczas liczenia średniej ważonej normalizacja oznaczała podzielenie licznika i mianownika przez tą samą wartość). Macierze podobień-

stwa dla maksymalnej długości ścieżek równej 9 były w 100% wypełnione, a mimo tego spisywały się gorzej od swoich odpowiedników o maksymalnej długości ścieżek równej 5, w których znalazły się nieokreślone pola. Wartość parametru α nie miała większego znaczenie – macierze podobieństwa faktycznie się różniły, ale już same prognozy wychodziły niemal identyczne. Tak jak poprzednio normalizacja rankingów zwiększała jakość przewidywań (patrz rysunek 3.4d), natomiast ciężko jest stwierdzić, czy lepsze rezultaty dawało posługiwanie się podobieństwami kursów, czy studentów – w zależności od pozostałych parametrów raz jedne, raz drugie okazywały się skuteczniejsze. Mimo, że metody grafowe przewidywały nieco gorzej od innych, to warte zauważenia jest to, że zawsze były w stanie jakąś prognozę sporządzić.

Prognozy oparte o odległość euklidesową również zachowują się nieprzewidywalnie. Jedyna prawidłowość, którą da się zaobserwować to taka, że jeśli przy liczeniu odległości nie dokonano normalizacji, to normalizacja podczas liczenia prognozy zdecydowanie pogarsza wyniki. Poza tym przewidywanie używające odległości euklidesowej nie są zbyt dobre w porównaniu do innych metod.

Kurczenie podobieństw



Rysunek 3.5: RMSE w zależności od parametru β dla wybranych metod najbliższych sąsiadów z kurczeniem wag. Im większy jest parametr β tym istotniejszą rolę odgrywają podobieństwa oparte o dużą liczbę wspólnych kursów/studentów. Podobieństwa studentów zazwyczaj są oparte o dosyć sporą liczbę kursów i w ich przypadku kurczenie pomaga. Dla kursów podobieństwa najczęściej są oparte o małą ilość studentów i wtedy kurczenie przeszkadza.

W przypadku kurczenia podobieństw ciężko od jednoznaczną odpowiedź na pytanie czy polepsza ono jakość prognoz, czy wręcz przeciwnie. Wybrano dwie metody z rodziny

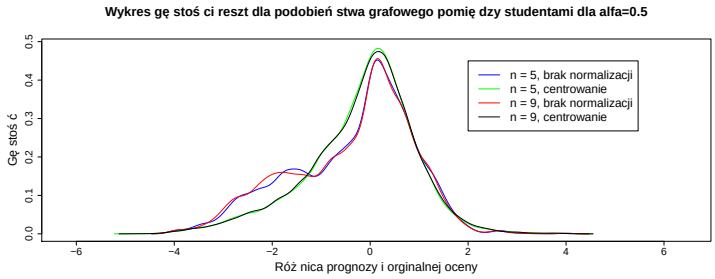
algorytmów najbliższych sąsiadów: kosinus kursów z centrowaniem dla prognoz i korelacja Pearsona studentów z normalizacją z-score dla prognoz, na których badano wpływ kurczenia podobieństw. Okazało się, że w przypadku studentów kurczenie podobieństw poprawiało trafność prognoz, natomiast w przypadku kursów pogarszało (na rysunku 3.5 można to zobaczyć dla wybranych parametrów k, ale dla pozostałych zależność jest podobna). Wyjaśnienie tego faktu znajduje się najprawdopodobniej w poprzedniej sekcji. Znajdują się tam dane na temat tego na podstawie ilu wspólnych rankingów obliczano podobieństwo. Ponieważ dla większości par kursów jest to bardzo mała wartość, to skurczenie wag nic nie wnosi, natomiast dla studentów są to wartości większe i tam już może pomóc.

Obserwacje

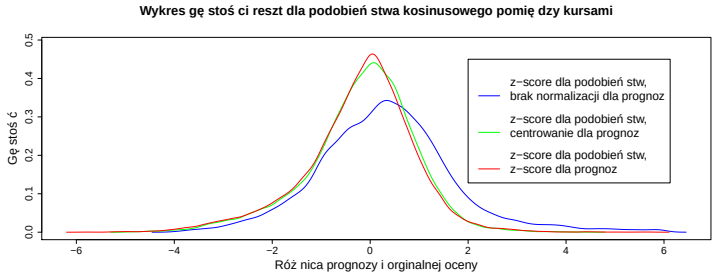
W rozdziale 3.3.3 zaproponowano kilka dodatkowych sposobów weryfikacji i oceny algorytmów prognozujących. W niniejszym rozdziale znajdują się najciekawsze obserwacje dotyczące metod najbliższych sąsiadów.

Zwróćmy uwagę na rysunek 3.6, który przedstawia rozkład różnic pomiędzy prognozą, a orginalną oceną dla wybranych algorytmów. Na ilustracji 3.6a widzimy, jak rozkładają się reszty dla podobieństwa grafowego pomiędzy studentami. Wynika z niego, że w przypadku braku normalizacji rankingów użytych do wygenerowania prognoz metoda ta ma tendencję do proponowania zbyt niskich ocen. Odwrotna sytuacja ma miejsca w przypadku kosinusowego podobieństwa między kursami – tutaj w przypadku braku normalizacji dla prognoz algorytm wydaje się sugerować zbyt wysokie rankingi. Odległość euklidesowa wydaje się być pomiędzy dwoma wcześniejszymi przypadkami, bo brak normalizacji dla prognoz oznacza, że część rankingów przewiduje za wysoko, część za nisko, ale żadna z tych tendencji nie jest dominująca.

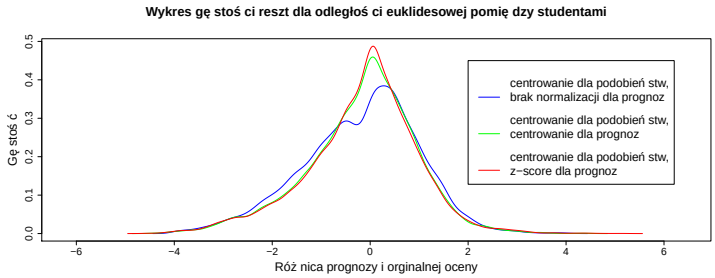
Kolejny obrazek (rysunek 3.7) przedstawia mapy podobieństwa pomiędzy wybranymi kursami. Kursy zostały dobrane tak, żeby osoba mająca styczność z matematyką na poziomie uniwersyteckim mogła ocenić, które metoda wyznaczyła podobieństwo najbardziej zbliżone do tego rzeczywistego (czegokolwiek byśmy nie rozumieli pod pojęciem podobieństwa kursów). Na rysunku 3.7d wyraźnie widać, że metoda grafowa bierze pod uwagę jedynie to na jakie przedmioty student uczęszczał, a nie to jakie uzyskał oceny – przedmioty, które nie dla wszystkich były obowiązkowe (Algebra 2, Statystyka 1 i Topologia 2) zostały ocenione jako mało podobne do innych, natomiast pozostałe kursy są między sobą bardzo podobne. Sensowniej wyglądają podobieństwa dla kosinusa i korelacji Pearsona (rysunki 3.7b i 3.7c). Warto zwrócić uwagę, że dla korelacji Pearsona mniejsza jest rozpiętość skali, stąd więcej tam zimnych barw, oznaczających podobieństwo poniżej środka skali. Widać jednak, że i te podobieństwa nie są idealne, bo np. podobieństwo Topologi 1 i Topologi 2 trafia w dolne rejony skali. W przypadku dostosowanego kosinusa (rysunek 3.7a) ciężko znaleźć jakąś regularność rządzącą tym, jak określone są podobieństwa przedmiotów– większość par przedmiotów ma podobieństwo powyżej środka skali, a te które mają to podobieństwo poniżej środka skali wydają się być wybrane losowo. Wszystko to przekłada się na trafność przewidywań. Jeżeli przy tworzeniu prognoz nie dokonamy żadnych normalizacji rankingów, to najlepiej wypadają



(a)



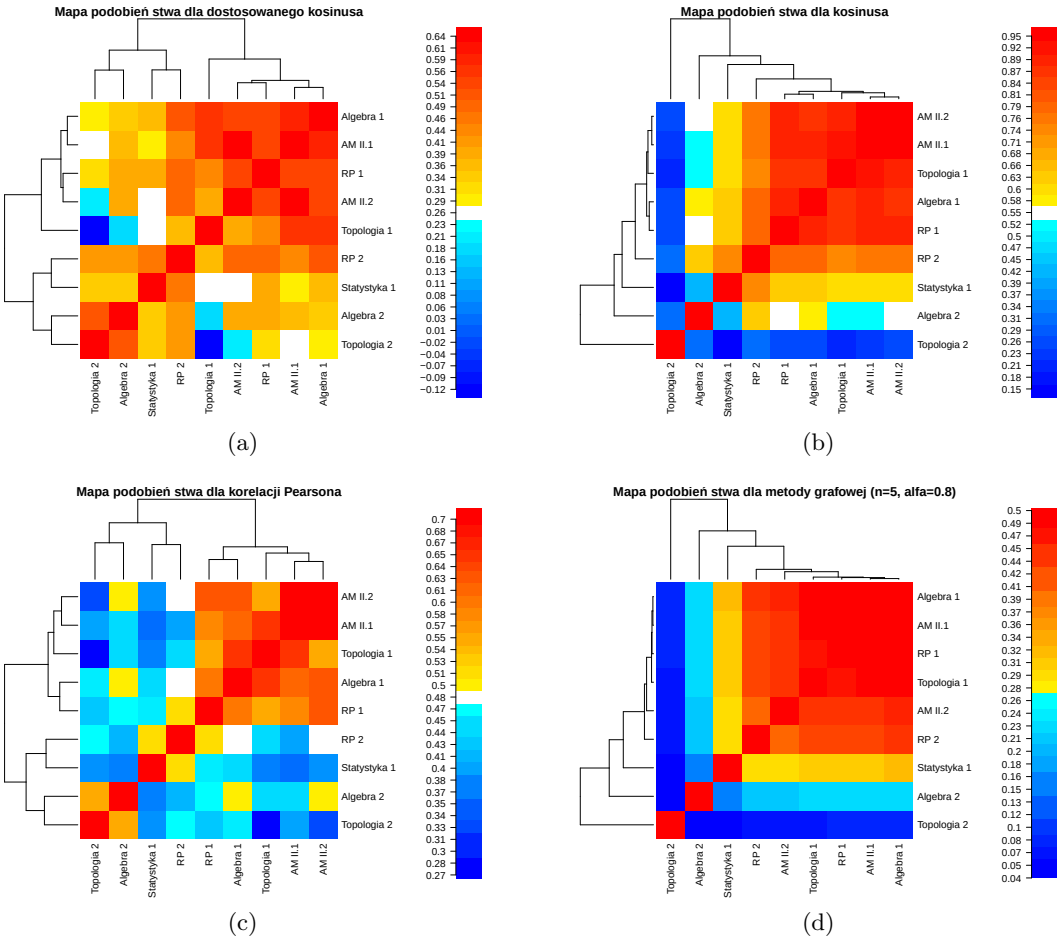
(b)



(c)

Rysunek 3.6: Gęstość różnic pomiędzy prognozą, a orginalną oceną dla wybranych algorytmów najbliższych sąsiadów. Każdy z rysunków odpowiada jednej metodzie liczenia podobieństwa: na (a) jest to podobieństwo grafowe studentów dla $\alpha = 0.5$, na (b) kosinus kursów, a na (c) odległość euklidesowa studentów. Z rysunków (b) i (c) widać, że jeśli przed policzeniem podobieństw dokonano normalizacji, to brak normalizacji podczas liczenia prognoz zdecydowanie pogarsza ich jakość. W przypadku podobieństwa grafowego normalizacja też była korzystna, bo niwelowała dziwny efekt widoczny na rysunku (a) (bardzo często prognoza była o 2 większa od prawdziwej oceny.)

te utworzone na podstawie kosinusa i korelacji Pearsona, nieco gorzej te pochodzące od podobieństw grafowych, a najsłabiej od dostosowanego kosinusa.



Rysunek 3.7: Mapy podobieństw wybranych przedmiotów i miar. Kolor komórki oznacza podobieństwo kursów z wiersza i kolumny, w których się znajduje. Kolory cieplejsze oznaczają większe podobieństwo, zimniejsze mniejsze, przy czym należy zwrócić uwagę, że skale na każdym z rysunków są różne. Na rysunku (a) znajduje się mapa podobieństwa dla dostosowanego kosinusa. Rysunek (b) obrazuje podobieństwo kosinusowe, natomiast (c) podobieństwo oparte o korelację Pearsona (chyba najlepiej wyłapujące zależności między przedmiotami). Na ilustracji (d) przedstawiono podobieństwo grafowe dla $n = 5$ oraz $\alpha = 0.8$.

Ocena systemów rekomendacyjnych z rodziny najbliższych sąsiadów

Wyniki przedstawione w poprzednich sekcjach, chociaż ciekawe, nie pokazują jak dobry jest system rekomendacyjny z punktu widzenia końcowego użytkownika. Tutaj postaramy się ocenić to jaką wartość mają rekomendację algorytmów najbliższych sąsiadów dla zainteresowanych nimi studentów. Na potrzeby tej sekcji zaokrąglamy prognozowane oceny do najbliższej oceny, jaką student mógł otrzymać (mamy zatem oceny od

1 do 5 z krokiem 0.5; prognozy, które nie zmieściły się w tym zakresie zaokrąglamy do bliższego końca skali). Będziemy oceniać jedynie wybrane algorytmy– trzy ocenione wcześniej jako dobre i jeden oceniony wcześniej jako nie najlepszy:

- podobieństwo kursów liczone za pomocą kosinusa, centrowanie dla podobieństw, centrowanie dla prognoz (dobry; oznaczany w tej sekcji przez KKCC);
- podobieństwo studentów liczone za pomocą kosinusa, brak normalizacji dla podobieństw, centrowanie dla prognoz (dobry; oznaczany w tej sekcji przez SKBC);
- podobieństwo studentów liczone za pomocą kosinusa, centrowanie dla podobieństw, centrowanie dla prognoz (dobry; oznaczany w tej sekcji przez SKCC);
- podobieństwo studentów liczone za pomocą korelacji Pearsona, brak normalizacji dla podobieństw, brak normalizacji dla prognoz (kiepski; oznaczany w tej sekcji przez SPBB).

Algorytm	KKCC	SKBC	SKCC	SPBB
Procent poprawnie przewidzianych ocen	22.83%	24.21%	**24.87%**	18.59%

Tablica 3.6: Odsetek poprawnie przewidzianych ocen dla wybranych algorytmów najbliższych sąsiadów. Najwyższa wartość w okolicach 25% nie jest wybitnym rezultatem.

Dla użytkownika systemu rekomendacyjnego najistotniejszą informacją jest to jak bardzo może mu ufać, a dokładniej jaka jest szansa, że przewidziane przez system oceny pokryją się z tymi rzeczywistymi. W tabeli 3.6 znajdują się informacje na temat tego, jaka część ocen w zbiorze testowym została dokładnie przewidziana. Dość ciekawy jest fakt, że różnica pomiędzy najgorszymi i najlepszymi algorytmami nie jest duża (około 6%). Niestety wynik w okolicach 25% w naszej ocenie jest wynikiem słabym.

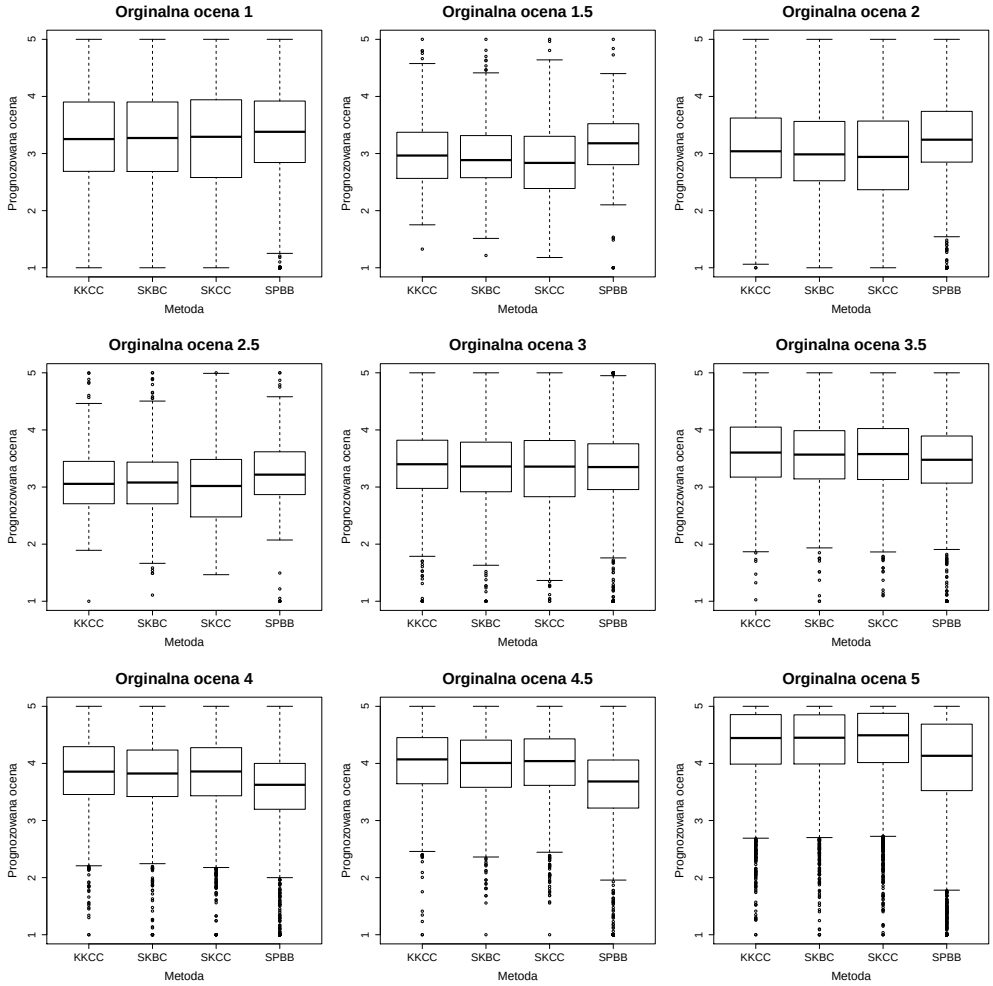
Dla części studentów nie jest jednak istotne to, czy system przewidział ich ocenę dokładnie, czy też pomylił się o pół oceny (o ile nie jest to połówka pomiędzy 2.5 a 3). Dla wielu studentów wystarczająca jest informacja, czy dostaną z danego kursu oceną wysoką (4.5–5), średnią (3–4), czy też niską (poniżej 3). Z drugiej strony duża ilość pomyłek polegających na złym zakwalifikowaniu oceny do grupy może być dla nich nie do przyjęcia – niska prognoza dla wysokiej oceny pewnie przejdzie niezauważona i nie będzie aż tak bolesna, ale średnia lub, co gorsza, wysoka prognoza dla niskiej oceny będzie w ich odczuciu dyskwalifikowała system rekomendacyjny (bo zachęci ich do uczęszczania na kurs, z którym będą mieli problemy). Oczywiście dla różnych studentów granice pomiędzy ocenami niskimi, średnimi i wysokimi przebiegają w różnych miejscach, jednak wydaje nam się, że większości zaproponowany podział będzie odpowiadał. Tabela 3.7 zawiera informacje na temat tego, jak dobrze algorytmy przewidują grupę do jakiej trafi ocena. Najlepsze algorytmy potrafią zrobić to w 56% przypadków. Wydawać by się mogło, że to już całkiem niezły wynik, ale należy wziąć pod uwagę to, że algorytmy te bardzo dobrze spisują się dla ocen z grupy średniej, przyzwoicie dla wysokich ocen i słabo dla niskich ocen. To jest dosyć poważna wada, ponieważ nie pozwala studentom

Algorytm	KKCC	SKBC	SKCC	SPBB
Przewidziana grupa				
Orginalna grupa: niskie				
brak prognozy	2.18%	0.65%	2.91%	2.91%
niskie	31.93%	34.05%	**37.26%**	18.94%
średnie	55.84%	55.43%	49.60%	68.19%
wysokie	10.06%	9.87%	10.24%	9.96%
Orginalna grupa: średnie				
brak prognozy	2.78%	0.38%	1.00%	0.98%
niskie	9.52%	10.85%	13.95%	13.91%
średnie	70.54%	73.52%	68.70%	**75.40%**
wysokie	17.16%	15.24%	16.36%	9.07%
Orginalna grupa: wysokie				
brak prognozy	8.51%	0.60%	1.27%	1.27%
niskie	2.14%	2.24%	2.78%	10.92%
średnie	37.50%	41.07%	37.79%	48.94%
wysokie	51.86%	56.08%	**58.15%**	38.87%
Ogólny odsetek dobrej prognozy grupy				
	53.03%	56.30%	**56.37%**	45.98%

Tablica 3.7: Procent poprawnie przewidzianej grupy oceny dla wybranych algorytmów najbliższych sąsiadów. Oceny zostały podzielone na trzy grupy: niskie (mniejsze od 3), średnie (3–4) oraz wysokie (powyżej 4). Tabela jest podzielona na cztery części. W pierwszej bierzemy pod uwagę jedynie oceny, które w rzeczywistości należą do grupy ocen niskich, w drugiej te z grupy ocen średnich, a w trzeciej z wysokich. Ostatnia część bierze pod uwagę wszystkie oceny. W każdej części mamy pokazane ile procent z branych pod uwagę ocen zostało zakwalifikowanych do jakiej grupy. Ostatnia część mówi o tym, w jakiej części przypadków udało się odgadnąć prawdziwą grupę oceny.

rozpoznać kursów, które będą im sprawiały trudności. Potwierdziło się również to, że algorytm SPBB jest kiepski – ma dobry wynik jedynie dla średnich ocen, ale wynika to prawdopodobnie z tego, że większość ocen klasyfikuje on jako średnie. Fakty te znajdują potwierdzenie na rysunku 3.8.

Kolejną cechą generowanych prognoz, która może zainteresować studentów jest umiejętność przewidzenia tego czy ocena będzie wyższa, czy niższa od jego dotychczasowej średniej. Jest to szczególnie istotne dla studentów walczących o stypendia naukowe, którzy chcieliby zminimalizować liczbę kursów, które będą im średnią zaniżać. W tabeli 3.8 znajduje się podsumowanie tego, jak algorytmy potrafią prognozować relację oceny do średniej studenta (patrz też rozdział 3.3.2). Wyniki powyżej 65% nie są już wynikami złymi, ale wciąż trudno nazwać je bardzo dobrymi. Ponownie algorytmy zazwyczaj lepiej radzą sobie z wyższymi ocenami (tymi powyżej średniej), chociaż różnica nie jest tutaj strasznie duża, a w przypadku metody SPBB mamy do czynienia z sytuacją wręcz



Rysunek 3.8: Rozkład prognoz w podziale na orginalne oceny dla wybranych metod najbliższych sąsiadów. Widać wyraźnie, że wszystkie algorytmy mają problem z prognozowaniem niskich ocen i zdecydowanie je zawyżają. Przewidywanie ocen wyższych od 3 idzie im już znacznie lepiej (zwłaszcza 3.5). Różnice pomiędzy najlepszymi algorytmami są niewielkie, natomiast w porównaniu do nich algorytm najsłabszy (SPBB) bardziej zawyża niskie oceny i zaniża wysokie.

przeciwną.

Ostatnim, trudniejszym w interpretacji, sposobem oceny algorytmów jest policzenie korelacji rang Spearmana prognoz i rzeczywistych ocen. Informacje na temat tej miary znajdują się w rozdziale 3.3.2. Z tabeli 3.9, zawierającej wyniki dla badanych algorytmów, widzimy, że istnieje jakaś monotoniczna zależność pomiędzy ocenami i prognozami. Zależność ta jest większa dla dobrych algorytmów i zdecydowanie mniejsza dla gorszego. Oczywiście kolejność prognoz i rzeczywistych ocen nie zawsze jest taka sama, jednak

Algorytm	KKCC	SKBC	SKCC	SPBB
Oceny poniżej średniej	64.07%	66.54%	**67.48%**	63.44%
Oceny powyżej średniej	68.22%	68.43%	**68.48%**	51.86%
Ogółem	66.48%	67.81%	**68.11%**	57.49%

Tablica 3.8: Odsetek poprawnie przewidzianej relacji oceny do średniej studenta dla wybranych algorytmów najbliższych sąsiadów. W pierwszym wierszu bierzemy pod uwagę oceny, które naprawdę były niższe od średniej studenta i patrzymy w ilu przypadkach algorytm odgadł, że ocena będzie poniżej średniej. W drugim wierszu mamy analogiczne wyniki, ale dla ocen, które w rzeczywistości są wyższe od średniej studenta. W ostatnim wierszu pod uwagę brane są wszystkie oceny. Dobre algorytmy zazwyczaj lepiej radzą sobie z ocenami powyżej średniej.

Algorytm	KKCC	KSBC	KSCC	PSBB
Korelacja rang Spearmana	0.40	**0.43**	0.40	0.28

Tablica 3.9: Korelacja rang Spearmana dla wybranych algorytmów najbliższych sąsiadów. Za pomocą korelacji rang Spearmana określamy, jak bardzo zaproponowana kolejność przedmiotów (od tego z najwyższą oceną do tego z najniższą) jest podobna do kolejności orginalnej, zatem im wyższa wartość tym lepiej.

ciężko jest określić, czy korelacja rang Spearmana na poziomie 0.4 to dużo, czy mało.

Katalog z wynikami

Pełne wyniki opisywane w poprzedniej sekcji można znaleźć na płycie DVD w katalogu "/wyniki/nn". W kolejnych podkatalogach znajdują się:

- macierze podobieństwa w katalogu "macierze podobieństwa";
- wyniki pomiarów jakości prognoz (RMSE, MAE, MSPA, SRCC) w katalogu "pomiary";
- prognozy dla elementów ze zbioru testowego w katalogu "prognozy".

3.5.3. Modelowanie

W tabeli 3.10 przedstawiono wyniki pomiarów jakości prognoz dla modelowania. Analiza RMSE wskazuje, że prognozy wygenerowane przy użyciu modelowania są przynajmniej tak samo dobre, jak te pochodzące od najlepszych algorytmów najbliższych sąsiadów i dużo lepsze od obliczonych przez algorytmy slope one. Dużą przewagą metod opisywanych w tej sekcji jest to, że dla każdej pary student-kurs są w stanie zaproponować jakąś prognozę. Dosyć ciekawy jest fakt, że modele bazowe są zazwyczaj lepsze od swoich odpowiedników w wersji SVD. Może to wynikać z faktu, że te pierwsze były dużo

typ	regularyzacja	f	RMSE	MAE	MSPA	SRCC
bazowy	0.001	–	1.08	0.91	**0.67**	**0.39**
bazowy	0.01	–	1.09	0.91	**0.67**	**0.39**
bazowy	0.1	–	1.08	**0.90**	**0.67**	**0.39**
bazowy	1	–	**1.07**	0.91	0.66	**0.39**
svd	0.001	10	1.12	0.92	0.66	0.36
svd	0.001	20	1.12	0.92	0.65	0.36
svd	0.001	40	1.12	0.92	0.65	0.36
svd	0.01	10	1.12	0.92	0.65	0.36
svd	0.01	20	1.12	0.92	0.65	0.37
svd	0.01	40	1.12	0.92	0.65	0.36
svd	0.1	10	1.11	0.92	0.66	0.37
svd	0.1	20	1.09	0.91	0.65	0.38
svd	0.1	40	1.10	0.92	0.65	0.37
svd	1	10	**1.07**	0.91	0.66	**0.39**
svd	1	20	1.08	0.91	0.66	**0.39**
svd	1	40	**1.07**	0.91	0.66	**0.39**

Tablica 3.10: Wyniki dla modelowania. Różnice pomiędzy metodami są niewielkie, najlepiej wypadają te z parametrem λ_2 równym 1, ale tak naprawdę sporo też zależy od kryterium za pomocą, którego oceniamy algorytm. Zaskakujące jest to, że modele bazowe konkurują z modelami SVD (część nawet zdecydowanie pokonują).

lepiej dopasowane – uczenie trwało dłużej i na końcu tego procesu algorytm minimalizujący robił bardzo małe kroki, natomiast w przypadku modeli SVD pod koniec uczenia wartość minimalizowanej funkcji zmniejszała się dużo bardziej. Okazuje się również, że lepsze rezultaty przynosiło użycie dużego parametru regularyzacyjnego λ, co jest dosyć zaskakujące, ponieważ w konkursie Netflix'a dobrym wyborem były wartości w okolicach 0.01.

Podobnie jak w rozdziale 3.5.2, tak i tutaj postaramy się ocenić to, na ile użyteczne są systemy rekomendacyjne oparte o modele. Przyjrzymy się dokładniej czterem wybranym modelom:

- bazowy z parametrem regularyzacyjnym równym 1 (oznaczany R1F0);
- SVD z parametrem regularyzacyjnym równym 0.1 i parametrem f równym 20 (oznaczany R01F20);
- SVD z parametrem regularyzacyjnym równym 1 i parametrem f równym 10 (oznaczany R1F10);
- SVD z parametrem regularyzacyjnym równym 0.001 i parametrem f równym 40(oznaczany R0001F40).

Ogólna trafność prognoz jest podobna do tej w algorytmach najbliższych sąsiadów i wynosi około 21%. Powyższe modele jednakowo często zawyżały ocenę, jak i ją zaniżały. Nieco inaczej kształtuje się trafność prognoz w poszczególnych grupach (patrz rozdział 3.5.2) – dla modelowania wśród niskich ocen jest ona na poziomie 20%, natomiast dla średnich ocen wynosi około 80%. Dla ocen wysokich trafność prognoz to około 50%, przy czym wyjątkiem jest model R0001F40, w którym jest to 44% (za to dla niskich ocen model ten ma skuteczność 30%). Nieco więcej informacji na temat tego, jak rozkładały się oceny można odczytać z rysunku 3.9.

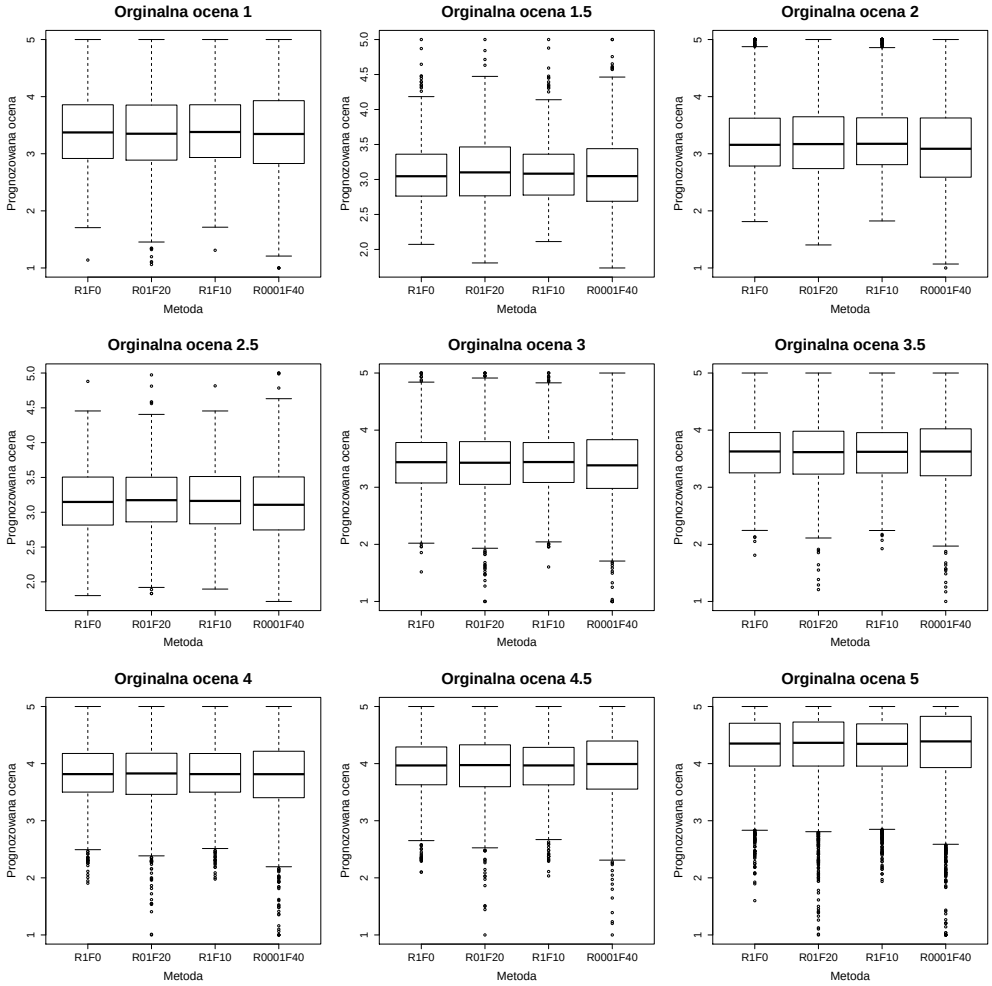
Analizowane modele w 66% procentach przypadków potrafiły ocenić po której stronie średniej znajdzie się przewidywana ocena, przy czym dla ocen które rzeczywiście były powyżej średniej było to 70%, a dla tych poniżej 61%. Korelacja rang w okolicach 0.38 była niższa od tej dla metod najbliższych sąsiadów.

RMSE sugeruje, że prognozy modeli są na podobnym poziomie, jak te pochodzące od algorytmów najbliższych sąsiadów, ale wobec powyższych faktów należy ocenić, że są one mniej użyteczne od tych przedstawionych w poprzednim rozdziale.

Katalog z wynikami

Pełne wyniki opisywane w poprzedniej sekcji można znaleźć na płycie DVD w katalogu "/wyniki/modelowanie". W kolejnych podkatalogach znajdują się:

- dopasowane parametry modeli w katalogu "dopasowane modele";
- wyniki pomiarów jakości prognoz (RMSE, MAE, MSPA, SRCC) w katalogu "pomiary";
- prognozy dla elementów ze zbioru testowego w katalogu "prognozy".



Rysunek 3.9: Rozkład prognoz w podziale na orginalne oceny dla wybranych modeli. Podobnie, jak dla algorytmów NN, tak i tutaj największe problemy stwarza prognozowanie niskich ocen (znów najsensowniej wyglądają prognozy dla 3.5 i 4). Różnice między dobrymi algorytmami są niewielkie, natomiast ten najsłabszy zazwyczaj medianę prognoz ma na tym samym poziomie, co te dobre, ale rozrzut prognoz jest u niego większy.

Rozdział 4

Podsumowanie

Celem niniejszej pracy była ocena tego, jak algorytmy i metody stosowane we współczesnych systemach rekomendacyjnych radzą sobie z rekomendowaniem kursów studentom. W pracy zaprezentowano zarys ogólnej teorii systemów rekomendacyjnych ze szczególnym uwzględnieniem metody slope one, algorytmu najbliższych sąsiadów oraz metod opartych o model. Przedstawiono również metody obliczeniowe wykorzystywane w systemach rekomendacyjnych, z których na szczególną uwagę zasługuje stochastyczna metoda najszybszego spadku.

Najistotniejszą częścią pracy było zastosowanie opisanych metod do danych pochodzących z bazy systemu USOS, przy czym prognozowano to, jaką ocenę student uzyska z danego kursu. Dane podzielono na zbiór treningowy i testowy. Metody uczono na zbiorze treningowym, a następnie sprawdzano jak radzą sobie na przykładach pochodzących ze zbioru testowego. Głównym kryterium oceny tego, jak daleko prognozy znajdują się od rzeczywistych ocen był pierwiastek błędu średniokwadratowego (RMSE). Niestety jakość przewidywań nie była zbyt wysoka, aczkolwiek w pewnych sytuacjach prognozy mogły być użyteczne (np. niska prognoza z dużym prawdopodobieństwem oznacza niską ocenę). Spośród zaprezentowanych metod najlepiej radziły sobie te, będące odmianami algorytmu najbliższych sąsiadów. Oczywiście teza ta nie dotyczy wszystkich zaprezentowanych algorytmów z tej rodziny, ale to wśród nich znalazły się metody, które oprócz niskiego RMSE, odznaczały się dość wysoką skutecznością w określaniu tego, czy ocena z danego kursu będzie wyższa, czy niższa od dotychczasowej średniej studenta oraz czy student zaliczy dany kurs, a proponowana przez nie kolejność przedmiotów (od tych z najwyższą oceną, do tych z najniższą) była najbardziej zbliżona do rzeczywistej. Gdyby należało wskazać jedną metodę, za pomocą której będą studentom rekomendowane kursy, to najlepsza wydaje się metoda najbliższych sąsiadów oparta o kosinusowe podobieństwo studentów z centrowaniem rankingów zarówno przed liczeniem podobieństw, jak i przed generowaniem prognoz.

Warto dodać, że pewnym odstępstwem od systemów rekomendacyjnych stosowanych w przemyśle jest to, że tutaj prognozujemy to jaką ocenę student otrzyma, a nie to, jak bardzo dany kurs będzie mu się podobał. W celu przewidywania tej drugiej wartości należałoby użyć danych z ankiet wypełnianych przez studentów, ale na chwilę obecną

jest ich zbyt mało nawet w stosunku do potrzeb systemów rekomendacyjnych.

Do przeprowadzenia analiz, obliczeń i wykonania rysunków wykorzystane zostało środowisko R. W przypadku tego ostatniego zadnia środowisko sprawdziło się bardzo dobrze Do pozostałych celów prawdopodobnie można znaleźć lepsze narzędzia. Język R jest interpretowany i w związku z tym dosyć wolny, natomiast w przypadku obliczeń wykonywanych w tej pracy nie były potrzebne żadne zaawansowane narzędzia statystyczne, które oferuje. Lepszym wyjściem mogło okazać się wykonanie obliczeń w jakimś szybszym języku kompilowanym (C, C++).

Wymienione wyżej mankamenty, to oczywiście pole do dalszego rozwoju tej pracy. Wykorzystanie danych z ankiet oznaczałoby zbliżenie się do prawdziwych systemów rekomendacyjnych (tzn. takich, w których sugerujemy użytkownikowi, to co będzie mu się podobało lub będzie dla niego interesujące). Wykorzystanie języka innego niż R pozwoliłoby na znaczne przyśpieszenie obliczeń, a poza tym mogłoby umożliwić integrację z USOS-em.

Dodatek A

Wyniki

Poniżej znajduje się pełne zestawienie wyników dla zbadanych metod. W przypadku algorytmów najbliższych sąsiadów nazwa metody oznaczona jest kodem, który należy interpretować w następujący sposób:

- pierwsza litera oznacza, czy podobieństwo było liczone pomiędzy kursami (K), czy studentami (S);
- przedostatnia litera oznacza normalizację użytą podczas liczenia podobieństw: brak normalizacji (B), centrowanie (C), normalizację z-score (Z);
- ostatnia litera oznacza normalizację użytą podczas liczenia prognozy: brak normalizacji (B), centrowanie (C), normalizację z-score (Z);
- pozostałe litery oznaczają miarę za pomocą której liczono podobieństwa:
  - K – kosinus;
  - Dk – dostosowany kosinus;
  - P – korelacja Pearsona;
  - E – odległość euklidesowa
  - Gx.y – podobieństwo liczone metodą grafową z parametrem n równym x i parametrem α równym y.

	slope one	ważony slope one	bi-polar slope one
RMSE	1.1493	1.1506	1.3371
MAE	0.9094	0.9095	0.9668
MSPA	0.6751	0.6764	0.5836
SRCC	0.4089	0.4071	0.3569

Tablica A.1: Wyniki dla algorytmu slope one

	-0.2	0	0.2	0.4	0.6	0.8
KDkBB	1.833	1.101	1.123	1.173	1.254	1.312
KKCB	1.890	1.132	1.081	1.086	1.425	1.613
KKZB	2.015	1.131	1.078	1.073	1.345	1.326
KKBB	1.145	1.145	1.125	1.093	1.072	1.141
KECB	1.186	1.186	1.309	1.327	1.340	1.352
KEZB	1.186	1.186	1.301	1.316	1.344	1.356
KEBB	1.161	1.161	1.273	1.272	1.287	1.300
KPBB	1.278	1.128	1.124	1.147	1.266	1.344
SKCB	2.024	1.191	1.166	1.168	1.022	0.786
SKZB	2.050	1.196	1.170	1.169	1.039	0.875
SKBB	1.222	1.222	1.217	1.224	1.282	1.318
SECB	1.220	1.220	1.223	1.261	1.272	1.285
SEZB	1.222	1.222	1.236	1.279	1.290	1.301
SEBB	1.178	1.178	1.154	1.179	1.192	1.207
SPBB	1.336	1.192	1.191	1.185	1.205	1.242

Tablica A.2: RMSE dla algorytmów najbliższych sąsiadów (wybór sąsiadów powyżej wartości progowej)

	-0.2	0	0.2	0.4	0.6	0.8
KDkBB	1.178	0.905	0.906	0.916	0.938	0.952
KKCB	1.172	0.947	0.905	0.874	0.969	1.043
KKZB	1.206	0.946	0.902	0.868	0.944	0.918
KKBB	0.958	0.958	0.941	0.921	0.912	0.934
KECB	0.966	0.966	0.996	0.993	0.989	0.988
KEZB	0.964	0.964	0.988	0.982	0.983	0.980
KEBB	0.950	0.950	0.970	0.955	0.950	0.949
KPBB	1.009	0.946	0.939	0.932	0.962	0.983
SKCB	1.207	0.950	0.926	0.901	0.798	0.642
SKZB	1.214	0.953	0.928	0.902	0.806	0.677
SKBB	0.971	0.971	0.967	0.962	0.978	0.982
SECB	0.967	0.967	0.968	0.983	0.984	0.985
SEZB	0.970	0.970	0.978	0.993	0.993	0.993
SEBB	0.949	0.949	0.934	0.936	0.937	0.940
SPBB	1.000	0.954	0.953	0.948	0.952	0.967

Tablica A.3: MAE dla algorytmów najbliższych sąsiadów (wybór sąsiadów powyżej wartości progowej)

	-0.2	0	0.2	0.4	0.6	0.8
KDkBB	0.570	0.656	0.656	0.656	0.638	0.632
KKCB	0.490	0.574	0.636	0.679	0.668	0.674
KKZB	0.478	0.582	0.639	0.681	0.661	0.685
KKBB	0.533	0.533	0.561	0.589	0.621	0.605
KECB	0.520	0.520	0.534	0.558	0.573	0.578
KEZB	0.527	0.527	0.545	0.573	0.580	0.590
KEBB	0.552	0.552	0.575	0.603	0.615	0.620
KPBB	0.514	0.593	0.601	0.613	0.591	0.582
SKCB	0.581	0.593	0.624	0.658	0.723	0.789
SKZB	0.583	0.596	0.626	0.660	0.729	0.806
SKBB	0.580	0.580	0.583	0.597	0.595	0.604
SECB	0.578	0.578	0.579	0.577	0.583	0.588
SEZB	0.580	0.580	0.578	0.580	0.584	0.588
SEBB	0.595	0.595	0.616	0.627	0.631	0.633
SPBB	0.585	0.588	0.591	0.602	0.607	0.605

Tablica A.4: MSPA dla algorytmów najbliższych sąsiadów (wybór sąsiadów powyżej wartości progowej)

	-0.2	0	0.2	0.4	0.6	0.8
KDkBB	0.252	0.354	0.345	0.314	0.271	0.183
KKCB	0.010	0.251	0.247	0.316	0.351	-0.004
KKZB	0.016	0.268	0.275	0.360	0.415	0.326
KKBB	0.199	0.199	0.191	0.206	0.233	0.249
KECB	0.182	0.182	0.100	0.116	0.123	0.128
KEZB	0.166	0.166	0.122	0.157	0.158	0.175
KEBB	0.277	0.277	0.175	0.196	0.202	0.180
KPBB	0.132	0.242	0.253	0.243	0.232	0.166
SKCB	0.276	0.383	0.380	0.381	0.428	0.580
SKZB	0.274	0.378	0.372	0.387	0.387	0.541
SKBB	0.368	0.368	0.367	0.356	0.320	0.336
SECB	0.354	0.354	0.345	0.289	0.272	0.277
SEZB	0.352	0.352	0.324	0.276	0.265	0.273
SEBB	0.385	0.385	0.377	0.332	0.327	0.326
SPBB	0.346	0.379	0.369	0.380	0.337	0.322

Tablica A.5: SRCC dla algorytmów najbliższych sąsiadów (wybór sąsiadów powyżej wartości progowej)

	5	10	15	20	25	30	35	40	45	50	60	70	80	90	100	150
KDkBB	1.345	1.841	2.431	2.705	2.982	3.213	3.411	3.613	3.805	3.956	4.153	4.230	4.256	4.263	4.266	4.266
KDkBC	1.159	1.158	1.178	1.196	1.209	1.218	1.225	1.231	1.235	1.237	1.239	1.240	1.240	1.240	1.240	1.240
KDkBZ	1.333	1.272	1.270	1.275	1.281	1.281	1.281	1.281	1.280	1.279	1.277	1.277	1.276	1.276	1.276	1.276
KKBB	1.168	1.151	1.148	1.147	1.146	1.146	1.146	1.145	1.145	1.145	1.145	1.145	1.145	1.145	1.145	1.145
KKBC	1.129	1.111	1.108	1.107	1.107	1.107	1.107	1.107	1.107	1.106	1.106	1.106	1.106	1.106	1.106	1.106
KKBZ	1.177	1.151	1.145	1.142	1.141	1.140	1.139	1.139	1.139	1.139	1.138	1.138	1.138	1.138	1.138	1.138
KKCB	1.190	1.240	1.344	1.396	1.460	1.522	1.583	1.647	1.714	1.773	1.864	1.900	1.916	1.921	1.923	1.924
KKCC	1.118	1.100	1.099	1.099	1.098	1.097	1.097	1.096	1.096	1.096	1.096	1.096	1.096	1.096	1.096	1.096
KKCZ	1.122	1.105	1.104	1.104	1.103	1.102	1.101	1.101	1.101	1.101	1.100	1.099	1.099	1.099	1.099	1.099
KKZB	1.201	1.270	1.385	1.451	1.529	1.600	1.678	1.756	1.833	1.901	2.013	2.061	2.083	2.091	2.094	2.094
KKZC	1.127	1.110	1.109	1.109	1.108	1.107	1.107	1.107	1.106	1.106	1.107	1.107	1.107	1.107	1.107	1.107
KKZZ	1.204	1.172	1.164	1.160	1.157	1.153	1.150	1.147	1.146	1.145	1.143	1.140	1.140	1.139	1.139	1.139
KEBB	1.249	1.200	1.184	1.176	1.171	1.168	1.166	1.164	1.163	1.162	1.161	1.161	1.161	1.161	1.161	1.161
KEBC	1.181	1.150	1.141	1.135	1.132	1.130	1.127	1.126	1.125	1.125	1.124	1.124	1.124	1.124	1.124	1.124
KEBZ	1.480	1.380	1.341	1.319	1.305	1.295	1.287	1.281	1.278	1.275	1.273	1.273	1.273	1.273	1.273	1.273
KECB	1.282	1.230	1.213	1.203	1.198	1.194	1.192	1.189	1.188	1.187	1.186	1.186	1.186	1.186	1.186	1.186
KECC	1.177	1.147	1.138	1.132	1.129	1.127	1.125	1.124	1.123	1.123	1.122	1.122	1.122	1.122	1.122	1.122
KECZ	1.185	1.155	1.146	1.140	1.136	1.134	1.132	1.131	1.130	1.129	1.129	1.128	1.128	1.128	1.128	1.128
KEZB	1.279	1.228	1.211	1.202	1.197	1.194	1.191	1.189	1.188	1.187	1.187	1.186	1.186	1.186	1.186	1.186
KEZC	1.185	1.156	1.147	1.142	1.138	1.136	1.134	1.133	1.133	1.132	1.131	1.131	1.131	1.131	1.131	1.131
KEZZ	1.343	1.291	1.269	1.256	1.247	1.241	1.236	1.232	1.231	1.229	1.228	1.227	1.227	1.227	1.227	1.227
KG5.05BB	1.288	1.245	1.225	1.211	1.204	1.199	1.197	1.194	1.193	1.192	1.191	1.191	1.191	1.191	1.191	1.191
KG5.05BC	1.220	1.182	1.167	1.160	1.157	1.155	1.154	1.154	1.153	1.153	1.152	1.152	1.152	1.152	1.152	1.152
KG5.2BB	1.291	1.247	1.227	1.214	1.206	1.202	1.200	1.198	1.196	1.196	1.195	1.195	1.195	1.195	1.195	1.195
KG5.2BC	1.222	1.183	1.168	1.162	1.159	1.157	1.156	1.156	1.156	1.155	1.155	1.155	1.155	1.154	1.154	1.154
KG5.5BB	1.291	1.247	1.227	1.214	1.207	1.202	1.200	1.198	1.197	1.196	1.195	1.195	1.195	1.195	1.195	1.195
KG5.5BC	1.222	1.183	1.168	1.162	1.159	1.157	1.156	1.156	1.156	1.155	1.155	1.155	1.155	1.155	1.155	1.155
KG5.8BB	1.291	1.247	1.227	1.214	1.207	1.202	1.200	1.198	1.197	1.196	1.195	1.195	1.195	1.195	1.195	1.195
KG5.8BC	1.222	1.183	1.168	1.162	1.159	1.157	1.156	1.156	1.156	1.155	1.155	1.155	1.155	1.155	1.155	1.155
KG9.05BB	1.299	1.261	1.244	1.235	1.228	1.224	1.221	1.219	1.218	1.218	1.217	1.217	1.217	1.217	1.217	1.217
KG9.05BC	1.228	1.191	1.180	1.177	1.173	1.171	1.171	1.171	1.171	1.171	1.170	1.170	1.170	1.170	1.170	1.170
KG9.2BB	1.299	1.261	1.244	1.235	1.228	1.224	1.221	1.220	1.219	1.218	1.217	1.217	1.217	1.217	1.217	1.217
KG9.2BC	1.228	1.191	1.180	1.177	1.173	1.171	1.171	1.171	1.171	1.171	1.170	1.170	1.170	1.170	1.170	1.170
KG9.5BB	1.299	1.261	1.244	1.235	1.228	1.224	1.221	1.220	1.219	1.218	1.217	1.217	1.217	1.217	1.217	1.217
KG9.5BC	1.228	1.191	1.180	1.177	1.173	1.171	1.171	1.171	1.171	1.171	1.170	1.170	1.170	1.170	1.170	1.170
KG9.8BB	1.299	1.261	1.244	1.235	1.228	1.224	1.221	1.220	1.219	1.218	1.217	1.217	1.217	1.217	1.217	1.217
KG9.8BC	1.228	1.191	1.180	1.177	1.173	1.171	1.171	1.171	1.171	1.171	1.170	1.170	1.170	1.170	1.170	1.170
KPBB	1.191	1.267	1.459	1.529	1.621	1.717	1.797	1.880	1.972	2.051	2.173	2.224	2.247	2.253	2.257	2.258
KPBC	1.127	1.104	1.100	1.100	1.099	1.099	1.100	1.101	1.101	1.101	1.101	1.101	1.100	1.100	1.100	1.100
KPBZ	1.128	1.105	1.103	1.102	1.102	1.102	1.103	1.103	1.103	1.103	1.102	1.100	1.100	1.100	1.100	1.100
SKBB	1.222	1.187	1.181	1.178	1.177	1.178	1.179	1.181	1.183	1.185	1.189	1.192	1.195	1.198	1.201	1.209
SKBC	1.114	1.090	1.085	1.082	1.080	1.080	1.080	1.080	1.081	1.081	1.082	1.082	1.083	1.083	1.084	1.085
SKBZ	1.116	1.093	1.087	1.084	1.082	1.081	1.081	1.081	1.081	1.081	1.082	1.082	1.083	1.083	1.083	1.084
SKCB	1.415	1.460	1.486	1.504	1.522	1.544	1.570	1.610	1.635	1.649	1.671	1.691	1.709	1.726	1.742	1.791
SKCC	1.124	1.104	1.097	1.093	1.090	1.087	1.086	1.085	1.084	1.084	1.083	1.082	1.082	1.082	1.081	1.080
SKCZ	1.130	1.110	1.101	1.097	1.094	1.090	1.089	1.087	1.087	1.086	1.085	1.084	1.084	1.083	1.083	1.081
SKZB	1.416	1.463	1.490	1.509	1.529	1.551	1.578	1.619	1.644	1.658	1.680	1.703	1.721	1.739	1.756	1.808
SKZC	1.128	1.108	1.101	1.097	1.094	1.091	1.089	1.089	1.088	1.088	1.087	1.087	1.086	1.086	1.085	1.084
SKZZ	1.138	1.116	1.107	1.103	1.099	1.096	1.094	1.092	1.092	1.091	1.090	1.090	1.089	1.089	1.088	1.087
SEBB	1.201	1.168	1.158	1.152	1.148	1.145	1.144	1.143	1.142	1.141	1.140	1.140	1.140	1.140	1.141	1.144
SEBC	1.164	1.133	1.121	1.114	1.110	1.106	1.104	1.102	1.101	1.099	1.098	1.096	1.095	1.094	1.093	1.090
SEBZ	1.173	1.139	1.125	1.116	1.111	1.107	1.105	1.103	1.100	1.099	1.097	1.095	1.094	1.093	1.092	1.090
SECB	1.268	1.237	1.226	1.220	1.216	1.214	1.213	1.212	1.211	1.211	1.210	1.209	1.209	1.209	1.209	1.210
SECC	1.163	1.136	1.124	1.117	1.112	1.109	1.107	1.105	1.104	1.103	1.101	1.099	1.098	1.097	1.096	1.093
SECZ	1.174	1.146	1.133	1.124	1.119	1.116	1.113	1.110	1.109	1.107	1.105	1.103	1.101	1.100	1.099	1.095
SEZB	1.275	1.242	1.229	1.223	1.220	1.218	1.216	1.215	1.214	1.213	1.212	1.211	1.211	1.211	1.211	1.211
SEZC	1.170	1.142	1.130	1.123	1.119	1.116	1.113	1.111	1.110	1.108	1.106	1.105	1.103	1.103	1.102	1.099
SEZZ	1.174	1.145	1.132	1.125	1.120	1.117	1.114	1.111	1.109	1.108	1.106	1.104	1.102	1.101	1.100	1.097
SG5.05BB	1.327	1.306	1.301	1.299	1.296	1.291	1.285	1.282	1.280	1.280	1.281	1.282	1.281	1.278	1.276	1.267
SG5.05BC	1.162	1.133	1.125	1.121	1.119	1.115	1.111	1.108	1.107	1.106	1.105	1.104	1.103	1.102	1.102	1.099
SG5.05BZ	1.162	1.133	1.125	1.121	1.119	1.115	1.111	1.108	1.107	1.106	1.105	1.104	1.103	1.102	1.102	1.099
SG5.2BB	1.328	1.307	1.302	1.299	1.297	1.292	1.287	1.283	1.281	1.281	1.283	1.283	1.282	1.279	1.277	1.269
SG5.2BC	1.164	1.134	1.125	1.121	1.119	1.115	1.111	1.109	1.107	1.106	1.105	1.104	1.103	1.103	1.102	1.099
SG5.2BZ	1.164	1.134	1.125	1.121	1.119	1.115	1.111	1.109	1.107	1.106	1.105	1.104	1.103	1.103	1.102	1.099
SG5.5BB	1.328	1.307	1.302	1.299	1.297	1.292	1.287	1.283	1.281	1.281	1.283	1.283	1.282	1.279	1.277	1.269
SG5.5BC	1.164	1.134	1.125	1.121	1.119	1.115	1.111	1.109	1.107	1.106	1.105	1.104	1.103	1.103	1.102	1.099
SG5.5BZ	1.164	1.134	1.125	1.121	1.119	1.115	1.111	1.109	1.107	1.106	1.105	1.104	1.103	1.103	1.102	1.099
SG5.8BB	1.328	1.307	1.302	1.299	1.297	1.292	1.287	1.283	1.281	1.281	1.283	1.283	1.282	1.279	1.277	1.269
SG5.8BC	1.164	1.134	1.125	1.121	1.119	1.115	1.111	1.109	1.107	1.106	1.105	1.104	1.103	1.103	1.102	1.099
SG5.8BZ	1.164	1.134	1.125	1.121	1.119	1.115	1.111	1.109	1.107	1.106	1.105	1.104	1.103	1.103	1.102	1.099
SG9.05BB	1.346	1.313	1.307	1.307	1.309	1.308	1.307	1.309	1.307	1.308	1.311	1.311	1.309	1.306	1.304	1.296
SG9.05BC	1.176	1.137	1.129	1.124	1.123	1.120	1.116	1.113	1.112	1.112	1.111	1.110	1.109	1.108	1.107	1.104
SG9.05BZ	1.176	1.137	1.129	1.124	1.123	1.120	1.116	1.113	1.112	1.112	1.111	1.110	1.109	1.108	1.107	1.104
SG9.2BB	1.346	1.313	1.306	1.306	1.309	1.308	1.307	1.309	1.307	1.308	1.311	1.311	1.309	1.306	1.304	1.296
SG9.2BC	1.176	1.137	1.129	1.124	1.123	1.120	1.116	1.113	1.112	1.112	1.111	1.110	1.109	1.108	1.107	1.104
SG9.2BZ	1.176	1.137	1.129	1.124	1.123	1.120	1.116	1.113	1.112	1.112	1.111	1.110	1.109	1.108	1.107	1.104
SG9.5BB	1.346	1.313	1.306	1.306	1.309	1.308	1.307	1.309	1.307	1.308	1.311	1.311	1.309	1.306	1.304	1.296
SG9.5BC	1.176	1.137	1.129	1.124	1.123	1.120	1.116	1.113	1.112	1.112	1.111	1.110	1.109	1.108	1.107	1.104
SG9.5BZ	1.176	1.137	1.129	1.124	1.123	1.120	1.116	1.113	1.112	1.112	1.111	1.110	1.109	1.108	1.107	1.104
SG9.8BB	1.346	1.313	1.306	1.306	1.309	1.308	1.307	1.309	1.307	1.308	1.311	1.311	1.309	1.306	1.304	1.296
SG9.8BC	1.176	1.137	1.129	1.124	1.123	1.120	1.116	1.113	1.112	1.112	1.111	1.110	1.109	1.108	1.107	1.104
SG9.8BZ	1.176	1.137	1.129	1.124	1.123	1.120	1.116	1.113	1.112	1.112	1.111	1.110	1.109	1.108	1.107	1.104
SPBB	1.453	1.501	1.535	1.557	1.579	1.608	1.638	1.687	1.724	1.740	1.768	1.794	1.815	1.841	1.863	1.924
SPBC	1.142	1.115	1.105	1.100	1.096	1.093	1.091	1.089	1.088	1.087	1.086	1.085	1.084	1.084	1.083	1.082
SPBZ	1.136	1.112	1.104	1.099	1.095	1.092	1.090	1.088	1.087	1.086	1.085	1.084	1.083	1.083	1.083	1.081

Tablica A.6: RMSE dla algorytmów najbliższych sąsiadów (wybór k najbliższych) cz. 1

	200	250	300	350	400	450	500	600	700	800	900	1000	1200	1400	1600	1800
KDkBB	4.266	4.266	4.266	4.266	4.266	4.266	4.266	4.266	4.266	4.266	4.266	4.266	4.266	4.266	4.266	4.266
KDkBC	1.240	1.240	1.240	1.240	1.240	1.240	1.240	1.240	1.240	1.240	1.240	1.240	1.240	1.240	1.240	1.240
KDkBZ	1.276	1.276	1.276	1.276	1.276	1.276	1.276	1.276	1.276	1.276	1.276	1.276	1.276	1.276	1.276	1.276
KKBB	1.145	1.145	1.145	1.145	1.145	1.145	1.145	1.145	1.145	1.145	1.145	1.145	1.145	1.145	1.145	1.145
KKBC	1.106	1.106	1.106	1.106	1.106	1.106	1.106	1.106	1.106	1.106	1.106	1.106	1.106	1.106	1.106	1.106
KKBZ	1.138	1.138	1.138	1.138	1.138	1.138	1.138	1.138	1.138	1.138	1.138	1.138	1.138	1.138	1.138	1.138
KKCB	1.924	1.924	1.924	1.924	1.924	1.924	1.924	1.924	1.924	1.924	1.924	1.924	1.924	1.924	1.924	1.924
KKCC	1.096	1.096	1.096	1.096	1.096	1.096	1.096	1.096	1.096	1.096	1.096	1.096	1.096	1.096	1.096	1.096
KKCZ	1.099	1.099	1.099	1.099	1.099	1.099	1.099	1.099	1.099	1.099	1.099	1.099	1.099	1.099	1.099	1.099
KKZB	2.094	2.094	2.094	2.094	2.094	2.094	2.094	2.094	2.094	2.094	2.094	2.094	2.094	2.094	2.094	2.094
KKZC	1.107	1.107	1.107	1.107	1.107	1.107	1.107	1.107	1.107	1.107	1.107	1.107	1.107	1.107	1.107	1.107
KKZZ	1.139	1.139	1.139	1.139	1.139	1.139	1.139	1.139	1.139	1.139	1.139	1.139	1.139	1.139	1.139	1.139
KEBB	1.161	1.161	1.161	1.161	1.161	1.161	1.161	1.161	1.161	1.161	1.161	1.161	1.161	1.161	1.161	1.161
KEBC	1.124	1.124	1.124	1.124	1.124	1.124	1.124	1.124	1.124	1.124	1.124	1.124	1.124	1.124	1.124	1.124
KEBZ	1.273	1.273	1.273	1.273	1.273	1.273	1.273	1.273	1.273	1.273	1.273	1.273	1.273	1.273	1.273	1.273
KECB	1.186	1.186	1.186	1.186	1.186	1.186	1.186	1.186	1.186	1.186	1.186	1.186	1.186	1.186	1.186	1.186
KECC	1.122	1.122	1.122	1.122	1.122	1.122	1.122	1.122	1.122	1.122	1.122	1.122	1.122	1.122	1.122	1.122
KECZ	1.128	1.128	1.128	1.128	1.128	1.128	1.128	1.128	1.128	1.128	1.128	1.128	1.128	1.128	1.128	1.128
KEZB	1.186	1.186	1.186	1.186	1.186	1.186	1.186	1.186	1.186	1.186	1.186	1.186	1.186	1.186	1.186	1.186
KEZC	1.131	1.131	1.131	1.131	1.131	1.131	1.131	1.131	1.131	1.131	1.131	1.131	1.131	1.131	1.131	1.131
KEZZ	1.227	1.227	1.227	1.227	1.227	1.227	1.227	1.227	1.227	1.227	1.227	1.227	1.227	1.227	1.227	1.227
KG5.05BB	1.191	1.191	1.191	1.191	1.191	1.191	1.191	1.191	1.191	1.191	1.191	1.191	1.191	1.191	1.191	1.191
KG5.05BC	1.152	1.152	1.152	1.152	1.152	1.152	1.152	1.152	1.152	1.152	1.152	1.152	1.152	1.152	1.152	1.152
KG5.2BB	1.195	1.195	1.195	1.195	1.195	1.195	1.195	1.195	1.195	1.195	1.195	1.195	1.195	1.195	1.195	1.195
KG5.2BC	1.154	1.154	1.154	1.154	1.154	1.154	1.154	1.154	1.154	1.154	1.154	1.154	1.154	1.154	1.154	1.154
KG5.5BB	1.195	1.195	1.195	1.195	1.195	1.195	1.195	1.195	1.195	1.195	1.195	1.195	1.195	1.195	1.195	1.195
KG5.5BC	1.155	1.155	1.155	1.155	1.155	1.155	1.155	1.155	1.155	1.155	1.155	1.155	1.155	1.155	1.155	1.155
KG5.8BB	1.195	1.195	1.195	1.195	1.195	1.195	1.195	1.195	1.195	1.195	1.195	1.195	1.195	1.195	1.195	1.195
KG5.8BC	1.155	1.155	1.155	1.155	1.155	1.155	1.155	1.155	1.155	1.155	1.155	1.155	1.155	1.155	1.155	1.155
KG9.05BB	1.217	1.217	1.217	1.217	1.217	1.217	1.217	1.217	1.217	1.217	1.217	1.217	1.217	1.217	1.217	1.217
KG9.05BC	1.170	1.170	1.170	1.170	1.170	1.170	1.170	1.170	1.170	1.170	1.170	1.170	1.170	1.170	1.170	1.170
KG9.2BB	1.217	1.217	1.217	1.217	1.217	1.217	1.217	1.217	1.217	1.217	1.217	1.217	1.217	1.217	1.217	1.217
KG9.2BC	1.170	1.170	1.170	1.170	1.170	1.170	1.170	1.170	1.170	1.170	1.170	1.170	1.170	1.170	1.170	1.170
KG9.5BB	1.217	1.217	1.217	1.217	1.217	1.217	1.217	1.217	1.217	1.217	1.217	1.217	1.217	1.217	1.217	1.217
KG9.5BC	1.170	1.170	1.170	1.170	1.170	1.170	1.170	1.170	1.170	1.170	1.170	1.170	1.170	1.170	1.170	1.170
KG9.8BB	1.217	1.217	1.217	1.217	1.217	1.217	1.217	1.217	1.217	1.217	1.217	1.217	1.217	1.217	1.217	1.217
KG9.8BC	1.170	1.170	1.170	1.170	1.170	1.170	1.170	1.170	1.170	1.170	1.170	1.170	1.170	1.170	1.170	1.170
KPBB	2.258	2.258	2.258	2.258	2.258	2.258	2.258	2.258	2.258	2.258	2.258	2.258	2.258	2.258	2.258	2.258
KPBC	1.100	1.100	1.100	1.100	1.100	1.100	1.100	1.100	1.100	1.100	1.100	1.100	1.100	1.100	1.100	1.100
KPBZ	1.100	1.100	1.100	1.100	1.100	1.100	1.100	1.100	1.100	1.100	1.100	1.100	1.100	1.100	1.100	1.100
SKBB	1.212	1.214	1.215	1.217	1.218	1.218	1.219	1.220	1.221	1.222	1.222	1.222	1.222	1.222	1.222	1.222
SKBC	1.085	1.085	1.085	1.085	1.085	1.085	1.085	1.085	1.085	1.085	1.085	1.085	1.084	1.084	1.084	1.084
SKBZ	1.084	1.084	1.084	1.083	1.083	1.083	1.083	1.083	1.082	1.082	1.082	1.082	1.082	1.082	1.082	1.082
SKCB	1.849	1.884	1.895	1.904	1.915	1.925	1.932	1.953	1.980	2.016	2.043	2.069	2.090	2.092	2.097	2.099
SKCC	1.080	1.079	1.079	1.079	1.079	1.080	1.080	1.080	1.081	1.082	1.083	1.085	1.087	1.088	1.088	1.088
SKCZ	1.081	1.080	1.080	1.080	1.080	1.080	1.080	1.080	1.081	1.081	1.082	1.084	1.086	1.087	1.087	1.087
SKZB	1.866	1.907	1.918	1.928	1.940	1.950	1.958	1.981	2.008	2.045	2.074	2.100	2.123	2.125	2.130	2.133
SKZC	1.084	1.085	1.084	1.085	1.085	1.085	1.085	1.086	1.087	1.088	1.089	1.091	1.094	1.095	1.095	1.095
SKZZ	1.086	1.086	1.086	1.086	1.086	1.086	1.086	1.086	1.087	1.088	1.089	1.091	1.093	1.094	1.095	1.095
SEBB	1.148	1.151	1.152	1.154	1.156	1.157	1.160	1.163	1.167	1.171	1.174	1.176	1.177	1.178	1.178	1.178
SEBC	1.089	1.088	1.087	1.087	1.087	1.086	1.086	1.086	1.086	1.086	1.087	1.087	1.087	1.087	1.087	1.087
SEBZ	1.089	1.088	1.087	1.087	1.086	1.086	1.086	1.086	1.086	1.086	1.086	1.087	1.087	1.087	1.087	1.087
SECB	1.212	1.213	1.214	1.214	1.215	1.216	1.217	1.218	1.219	1.220	1.220	1.220	1.220	1.220	1.220	1.220
SECC	1.092	1.091	1.090	1.090	1.090	1.090	1.090	1.090	1.090	1.090	1.090	1.090	1.090	1.090	1.090	1.090
SECZ	1.094	1.092	1.092	1.091	1.090	1.090	1.090	1.090	1.090	1.090	1.090	1.090	1.090	1.090	1.090	1.090
SEZB	1.213	1.214	1.214	1.214	1.215	1.216	1.216	1.218	1.219	1.221	1.222	1.222	1.222	1.222	1.222	1.222
SEZC	1.098	1.097	1.097	1.096	1.096	1.096	1.096	1.096	1.096	1.096	1.096	1.096	1.096	1.096	1.096	1.096
SEZZ	1.096	1.095	1.094	1.093	1.093	1.093	1.093	1.092	1.093	1.093	1.093	1.093	1.093	1.093	1.093	1.093
SG5.05BB	1.261	1.258	1.256	1.255	1.255	1.254	1.254	1.252	1.250	1.247	1.245	1.243	1.242	1.241	1.241	1.241
SG5.05BC	1.096	1.095	1.094	1.093	1.093	1.093	1.092	1.092	1.092	1.091	1.091	1.090	1.090	1.090	1.090	1.090
SG5.05BZ	1.096	1.095	1.094	1.093	1.093	1.093	1.092	1.092	1.092	1.091	1.091	1.090	1.090	1.090	1.090	1.090
SG5.2BB	1.263	1.259	1.257	1.256	1.256	1.255	1.255	1.253	1.250	1.248	1.245	1.243	1.242	1.241	1.241	1.241
SG5.2BC	1.097	1.095	1.094	1.094	1.093	1.093	1.093	1.092	1.092	1.091	1.091	1.090	1.090	1.090	1.090	1.090
SG5.2BZ	1.097	1.095	1.094	1.094	1.093	1.093	1.093	1.092	1.092	1.091	1.091	1.090	1.090	1.090	1.090	1.090
SG5.5BB	1.263	1.259	1.257	1.256	1.256	1.255	1.255	1.253	1.250	1.248	1.245	1.243	1.242	1.242	1.241	1.241
SG5.5BC	1.097	1.095	1.094	1.094	1.093	1.093	1.093	1.092	1.092	1.091	1.091	1.090	1.090	1.090	1.090	1.090
SG5.5BZ	1.097	1.095	1.094	1.094	1.093	1.093	1.093	1.092	1.092	1.091	1.091	1.090	1.090	1.090	1.090	1.090
SG5.8BB	1.263	1.259	1.257	1.256	1.256	1.255	1.255	1.253	1.250	1.248	1.245	1.243	1.242	1.242	1.241	1.241
SG5.8BC	1.097	1.095	1.094	1.094	1.093	1.093	1.093	1.092	1.092	1.091	1.091	1.090	1.090	1.090	1.090	1.090
SG5.8BZ	1.097	1.095	1.094	1.094	1.093	1.093	1.093	1.092	1.092	1.091	1.091	1.090	1.090	1.090	1.090	1.090
SG9.05BB	1.290	1.281	1.275	1.271	1.269	1.266	1.264	1.262	1.258	1.255	1.253	1.251	1.249	1.249	1.249	1.249
SG9.05BC	1.102	1.100	1.098	1.097	1.097	1.096	1.095	1.095	1.094	1.093	1.093	1.092	1.092	1.092	1.092	1.092
SG9.05BZ	1.102	1.100	1.098	1.097	1.097	1.096	1.095	1.095	1.094	1.093	1.093	1.092	1.092	1.092	1.092	1.092
SG9.2BB	1.290	1.281	1.275	1.271	1.269	1.266	1.264	1.262	1.258	1.255	1.253	1.251	1.250	1.249	1.249	1.249
SG9.2BC	1.102	1.100	1.098	1.097	1.097	1.096	1.095	1.095	1.094	1.093	1.093	1.092	1.092	1.092	1.092	1.092
SG9.2BZ	1.102	1.100	1.098	1.097	1.097	1.096	1.095	1.095	1.094	1.093	1.093	1.092	1.092	1.092	1.092	1.092
SG9.5BB	1.290	1.281	1.275	1.271	1.269	1.266	1.264	1.262	1.258	1.255	1.253	1.251	1.250	1.249	1.249	1.249
SG9.5BC	1.102	1.100	1.098	1.097	1.097	1.096	1.095	1.095	1.094	1.093	1.093	1.092	1.092	1.092	1.092	1.092
SG9.5BZ	1.102	1.100	1.098	1.097	1.097	1.096	1.095	1.095	1.094	1.093	1.093	1.092	1.092	1.092	1.092	1.092
SG9.8BB	1.290	1.281	1.275	1.271	1.269	1.266	1.264	1.262	1.258	1.255	1.253	1.251	1.250	1.249	1.249	1.249
SG9.8BC	1.102	1.100	1.098	1.097	1.097	1.096	1.095	1.095	1.094	1.093	1.093	1.092	1.092	1.092	1.092	1.092
SG9.8BZ	1.102	1.100	1.098	1.097	1.097	1.096	1.095	1.095	1.094	1.093	1.093	1.092	1.092	1.092	1.092	1.092
SPBB	1.994	2.037	2.052	2.064	2.078	2.089	2.098	2.125	2.158	2.203	2.241	2.274	2.297	2.301	2.308	2.310
SPBC	1.081	1.081	1.081	1.081	1.081	1.081	1.081	1.082	1.082	1.084	1.085	1.088	1.090	1.091	1.091	1.091
SPBZ	1.081	1.080	1.080	1.080	1.080	1.080	1.080	1.081	1.081	1.082	1.084	1.086	1.088	1.089	1.089	1.089

Tablica A.7: RMSE dla algorytmów najbliższych sąsiadów (wybór k najbliższych) cz. 2

	5	10	15	20	25	30	35	40	45	50	60	70	80	90	100	150
KDkBB	0.958	1.115	1.291	1.396	1.501	1.589	1.668	1.745	1.814	1.867	1.933	1.958	1.967	1.969	1.970	1.970
KDkBC	0.925	0.927	0.938	0.946	0.953	0.959	0.963	0.967	0.969	0.971	0.973	0.973	0.974	0.974	0.974	0.974
KDkBZ	0.985	0.966	0.968	0.971	0.974	0.975	0.976	0.976	0.977	0.978	0.979	0.980	0.980	0.980	0.980	0.980
KKBB	0.955	0.955	0.956	0.957	0.958	0.958	0.958	0.958	0.958	0.958	0.958	0.958	0.958	0.958	0.958	0.958
KKBC	0.922	0.915	0.914	0.913	0.913	0.913	0.913	0.913	0.912	0.912	0.912	0.912	0.912	0.912	0.912	0.912
KKBZ	0.931	0.921	0.919	0.918	0.918	0.917	0.917	0.916	0.916	0.916	0.916	0.916	0.916	0.916	0.916	0.916
KKCB	0.953	0.968	0.995	1.010	1.028	1.047	1.066	1.086	1.108	1.128	1.160	1.173	1.178	1.180	1.181	1.181
KKCC	0.917	0.910	0.909	0.909	0.909	0.908	0.908	0.908	0.908	0.908	0.908	0.908	0.908	0.908	0.908	0.908
KKCZ	0.911	0.904	0.904	0.904	0.904	0.904	0.904	0.904	0.904	0.904	0.904	0.904	0.905	0.905	0.905	0.905
KKZB	0.953	0.972	1.002	1.021	1.042	1.065	1.088	1.112	1.138	1.161	1.199	1.215	1.223	1.225	1.226	1.227
KKZC	0.919	0.912	0.911	0.911	0.910	0.910	0.910	0.909	0.909	0.909	0.909	0.909	0.909	0.910	0.910	0.910
KKZZ	0.936	0.925	0.922	0.921	0.920	0.918	0.918	0.917	0.917	0.917	0.917	0.916	0.916	0.916	0.916	0.916
KEBB	0.964	0.952	0.950	0.949	0.949	0.949	0.949	0.949	0.949	0.949	0.949	0.949	0.950	0.950	0.950	0.950
KEBC	0.941	0.929	0.925	0.923	0.922	0.921	0.920	0.919	0.919	0.919	0.919	0.919	0.919	0.919	0.919	0.919
KEBZ	1.037	1.006	0.993	0.985	0.980	0.976	0.973	0.971	0.970	0.969	0.968	0.968	0.968	0.968	0.968	0.968
KECB	0.988	0.975	0.971	0.969	0.968	0.967	0.967	0.966	0.966	0.966	0.966	0.966	0.966	0.966	0.966	0.966
KECC	0.942	0.931	0.927	0.925	0.923	0.923	0.922	0.922	0.921	0.921	0.921	0.921	0.921	0.921	0.921	0.921
KECZ	0.939	0.928	0.924	0.922	0.920	0.920	0.919	0.918	0.918	0.918	0.918	0.917	0.917	0.917	0.917	0.917
KEZB	0.983	0.971	0.968	0.966	0.965	0.965	0.964	0.964	0.964	0.964	0.964	0.964	0.964	0.964	0.964	0.964
KEZC	0.944	0.933	0.929	0.926	0.925	0.924	0.923	0.923	0.922	0.922	0.922	0.922	0.922	0.922	0.922	0.922
KEZZ	0.977	0.962	0.956	0.952	0.950	0.948	0.947	0.946	0.945	0.945	0.945	0.944	0.944	0.944	0.944	0.944
KG5.05BB	1.019	1.004	0.997	0.991	0.989	0.987	0.986	0.985	0.984	0.984	0.983	0.983	0.983	0.983	0.983	0.983
KG5.05BC	0.955	0.940	0.934	0.930	0.929	0.928	0.927	0.927	0.926	0.926	0.926	0.926	0.926	0.926	0.926	0.926
KG5.2BB	1.020	1.004	0.997	0.992	0.990	0.988	0.987	0.986	0.985	0.985	0.985	0.984	0.984	0.984	0.984	0.984
KG5.2BC	0.956	0.940	0.934	0.931	0.929	0.928	0.928	0.927	0.927	0.927	0.927	0.927	0.927	0.927	0.927	0.927
KG5.5BB	1.020	1.004	0.998	0.992	0.990	0.988	0.987	0.986	0.985	0.985	0.985	0.985	0.985	0.985	0.985	0.985
KG5.5BC	0.956	0.940	0.934	0.931	0.929	0.928	0.928	0.927	0.927	0.927	0.927	0.927	0.927	0.927	0.927	0.927
KG5.8BB	1.020	1.004	0.998	0.992	0.990	0.988	0.987	0.986	0.985	0.985	0.985	0.985	0.985	0.985	0.985	0.985
KG5.8BC	0.956	0.940	0.934	0.931	0.929	0.928	0.928	0.927	0.927	0.927	0.927	0.927	0.927	0.927	0.927	0.927
KG9.05BB	1.024	1.011	1.005	1.001	0.999	0.998	0.997	0.996	0.995	0.995	0.995	0.995	0.995	0.995	0.995	0.995
KG9.05BC	0.959	0.944	0.940	0.938	0.936	0.935	0.935	0.934	0.934	0.934	0.934	0.934	0.934	0.934	0.934	0.934
KG9.2BB	1.024	1.011	1.005	1.001	0.999	0.998	0.997	0.996	0.995	0.995	0.995	0.995	0.995	0.995	0.995	0.995
KG9.2BC	0.959	0.944	0.940	0.938	0.936	0.935	0.935	0.935	0.934	0.934	0.934	0.934	0.934	0.934	0.934	0.934
KG9.5BB	1.024	1.011	1.005	1.001	0.999	0.998	0.997	0.996	0.995	0.995	0.995	0.995	0.995	0.995	0.995	0.995
KG9.5BC	0.959	0.944	0.940	0.938	0.936	0.935	0.935	0.935	0.934	0.934	0.934	0.934	0.934	0.934	0.934	0.934
KG9.8BB	1.024	1.011	1.005	1.001	0.999	0.998	0.997	0.996	0.995	0.995	0.995	0.995	0.995	0.995	0.995	0.995
KG9.8BC	0.959	0.944	0.940	0.938	0.936	0.935	0.935	0.935	0.934	0.934	0.934	0.934	0.934	0.934	0.934	0.934
KPBB	0.946	0.968	1.020	1.043	1.074	1.109	1.137	1.166	1.200	1.230	1.275	1.294	1.302	1.304	1.305	1.306
KPBC	0.920	0.911	0.911	0.911	0.910	0.910	0.911	0.911	0.911	0.911	0.911	0.911	0.912	0.912	0.912	0.912
KPBZ	0.913	0.904	0.905	0.905	0.905	0.905	0.906	0.906	0.907	0.907	0.907	0.907	0.907	0.907	0.907	0.907
SKBB	0.953	0.944	0.944	0.944	0.944	0.946	0.947	0.948	0.950	0.951	0.953	0.955	0.957	0.958	0.959	0.963
SKBC	0.915	0.905	0.903	0.902	0.901	0.901	0.902	0.902	0.902	0.902	0.903	0.904	0.904	0.904	0.904	0.905
SKBZ	0.906	0.897	0.896	0.894	0.894	0.894	0.894	0.894	0.894	0.895	0.895	0.896	0.896	0.896	0.896	0.896
SKCB	0.968	0.986	0.997	1.005	1.012	1.020	1.030	1.045	1.054	1.059	1.068	1.076	1.084	1.091	1.097	1.116
SKCC	0.910	0.903	0.900	0.900	0.899	0.898	0.898	0.897	0.897	0.897	0.897	0.897	0.898	0.898	0.898	0.898
SKCZ	0.903	0.897	0.895	0.894	0.893	0.892	0.892	0.892	0.891	0.892	0.892	0.892	0.892	0.892	0.892	0.892
SKZB	0.969	0.987	0.999	1.007	1.015	1.023	1.033	1.048	1.057	1.063	1.072	1.080	1.088	1.095	1.102	1.122
SKZC	0.912	0.905	0.902	0.901	0.900	0.899	0.899	0.899	0.899	0.899	0.899	0.899	0.899	0.899	0.899	0.900
SKZZ	0.907	0.900	0.898	0.896	0.895	0.894	0.894	0.894	0.893	0.893	0.893	0.894	0.894	0.894	0.894	0.894
SEBB	0.938	0.930	0.927	0.926	0.925	0.925	0.925	0.925	0.925	0.925	0.925	0.926	0.926	0.927	0.928	0.930
SEBC	0.933	0.920	0.916	0.913	0.911	0.909	0.909	0.908	0.907	0.907	0.906	0.905	0.905	0.904	0.904	0.903
SEBZ	0.928	0.915	0.910	0.906	0.904	0.902	0.901	0.900	0.899	0.899	0.898	0.897	0.897	0.896	0.896	0.895
SECB	0.976	0.967	0.965	0.963	0.963	0.962	0.962	0.962	0.962	0.962	0.962	0.962	0.962	0.962	0.962	0.963
SECC	0.929	0.918	0.914	0.912	0.910	0.909	0.908	0.908	0.908	0.907	0.906	0.906	0.906	0.905	0.905	0.904
SECZ	0.929	0.918	0.913	0.910	0.908	0.907	0.906	0.905	0.905	0.904	0.903	0.902	0.902	0.901	0.901	0.900
SEZB	0.983	0.974	0.970	0.969	0.968	0.967	0.967	0.967	0.966	0.966	0.966	0.965	0.965	0.965	0.965	0.966
SEZC	0.936	0.925	0.920	0.917	0.915	0.914	0.913	0.912	0.912	0.911	0.911	0.910	0.910	0.909	0.909	0.908
SEZZ	0.925	0.915	0.910	0.907	0.906	0.905	0.904	0.903	0.903	0.902	0.901	0.901	0.900	0.900	0.900	0.899
SG5.05BB	1.003	0.998	0.997	0.996	0.994	0.992	0.990	0.989	0.989	0.989	0.990	0.990	0.990	0.989	0.988	0.986
SG5.05BC	0.929	0.919	0.915	0.913	0.912	0.910	0.909	0.907	0.906	0.906	0.906	0.905	0.905	0.905	0.904	0.903
SG5.05BZ	0.929	0.919	0.915	0.913	0.912	0.910	0.909	0.907	0.906	0.906	0.906	0.905	0.905	0.905	0.904	0.903
SG5.2BB	1.003	0.999	0.998	0.996	0.995	0.993	0.991	0.990	0.989	0.989	0.990	0.991	0.990	0.989	0.989	0.986
SG5.2BC	0.929	0.919	0.916	0.913	0.912	0.910	0.909	0.908	0.907	0.906	0.906	0.906	0.905	0.905	0.905	0.903
SG5.2BZ	0.929	0.919	0.916	0.913	0.912	0.910	0.909	0.908	0.907	0.906	0.906	0.906	0.905	0.905	0.905	0.903
SG5.5BB	1.003	0.999	0.998	0.996	0.995	0.993	0.991	0.990	0.989	0.989	0.990	0.991	0.990	0.989	0.989	0.986
SG5.5BC	0.929	0.919	0.916	0.913	0.912	0.910	0.909	0.908	0.907	0.906	0.906	0.906	0.905	0.905	0.905	0.903
SG5.5BZ	0.929	0.919	0.916	0.913	0.912	0.910	0.909	0.908	0.907	0.906	0.906	0.906	0.905	0.905	0.905	0.903
SG5.8BB	1.003	0.999	0.998	0.996	0.995	0.993	0.991	0.990	0.989	0.989	0.990	0.991	0.990	0.989	0.989	0.986
SG5.8BC	0.929	0.919	0.916	0.913	0.912	0.910	0.909	0.908	0.907	0.906	0.906	0.906	0.905	0.905	0.905	0.903
SG5.8BZ	0.929	0.919	0.916	0.913	0.912	0.910	0.909	0.908	0.907	0.906	0.906	0.906	0.905	0.905	0.905	0.903
SG9.05BB	1.006	0.999	0.997	0.999	1.000	0.999	0.999	0.999	0.999	1.000	1.001	1.001	1.000	0.999	0.999	0.996
SG9.05BC	0.934	0.921	0.918	0.917	0.916	0.915	0.912	0.911	0.910	0.910	0.910	0.910	0.909	0.908	0.908	0.906
SG9.05BZ	0.934	0.921	0.918	0.917	0.916	0.915	0.912	0.911	0.910	0.910	0.910	0.910	0.909	0.908	0.908	0.906
SG9.2BB	1.006	0.999	0.997	0.999	1.000	0.999	0.999	0.999	0.999	1.000	1.001	1.001	1.000	0.999	0.999	0.996
SG9.2BC	0.934	0.921	0.918	0.917	0.916	0.915	0.912	0.911	0.910	0.910	0.910	0.910	0.909	0.908	0.908	0.906
SG9.2BZ	0.934	0.921	0.918	0.917	0.916	0.915	0.912	0.911	0.910	0.910	0.910	0.910	0.909	0.908	0.908	0.906
SG9.5BB	1.006	0.999	0.997	0.999	1.000	0.999	0.999	0.999	0.999	1.000	1.001	1.001	1.000	0.999	0.999	0.996
SG9.5BC	0.934	0.921	0.918	0.917	0.916	0.915	0.912	0.911	0.910	0.910	0.910	0.910	0.909	0.908	0.908	0.906
SG9.5BZ	0.934	0.921	0.918	0.917	0.916	0.915	0.912	0.911	0.910	0.910	0.910	0.910	0.909	0.908	0.908	0.906
SG9.8BB	1.006	0.999	0.997	0.999	1.000	0.999	0.999	0.999	0.999	1.000	1.001	1.001	1.000	0.999	0.999	0.996
SG9.8BC	0.934	0.921	0.918	0.917	0.916	0.915	0.912	0.911	0.910	0.910	0.910	0.910	0.909	0.908	0.908	0.906
SG9.8BZ	0.934	0.921	0.918	0.917	0.916	0.915	0.912	0.911	0.910	0.910	0.910	0.910	0.909	0.908	0.908	0.906
SPBB	1.002	1.015	1.026	1.034	1.042	1.052	1.063	1.080	1.094	1.099	1.110	1.120	1.129	1.140	1.149	1.173
SPBC	0.925	0.913	0.909	0.907	0.906	0.905	0.904	0.903	0.903	0.903	0.903	0.902	0.902	0.902	0.902	0.903
SPBZ	0.913	0.904	0.900	0.899	0.898	0.897	0.896	0.896	0.896	0.896	0.895	0.895	0.895	0.896	0.896	0.896

Tablica A.8: MAE dla algorytmów najbliższych sąsiadów (wybór k najbliższych) cz. 1

	200	250	300	350	400	450	500	600	700	800	900	1000	1200	1400	1600	1800
KDkBB	1.970	1.970	1.970	1.970	1.970	1.970	1.970	1.970	1.970	1.970	1.970	1.970	1.970	1.970	1.970	1.970
KDkBC	0.974	0.974	0.974	0.974	0.974	0.974	0.974	0.974	0.974	0.974	0.974	0.974	0.974	0.974	0.974	0.974
KDkBZ	0.980	0.980	0.980	0.980	0.980	0.980	0.980	0.980	0.980	0.980	0.980	0.980	0.980	0.980	0.980	0.980
KKBB	0.958	0.958	0.958	0.958	0.958	0.958	0.958	0.958	0.958	0.958	0.958	0.958	0.958	0.958	0.958	0.958
KKBC	0.912	0.912	0.912	0.912	0.912	0.912	0.912	0.912	0.912	0.912	0.912	0.912	0.912	0.912	0.912	0.912
KKBZ	0.916	0.916	0.916	0.916	0.916	0.916	0.916	0.916	0.916	0.916	0.916	0.916	0.916	0.916	0.916	0.916
KKCB	1.181	1.181	1.181	1.181	1.181	1.181	1.181	1.181	1.181	1.181	1.181	1.181	1.181	1.181	1.181	1.181
KKCC	0.908	0.908	0.908	0.908	0.908	0.908	0.908	0.908	0.908	0.908	0.908	0.908	0.908	0.908	0.908	0.908
KKCZ	0.905	0.905	0.905	0.905	0.905	0.905	0.905	0.905	0.905	0.905	0.905	0.905	0.905	0.905	0.905	0.905
KKZB	1.227	1.227	1.227	1.227	1.227	1.227	1.227	1.227	1.227	1.227	1.227	1.227	1.227	1.227	1.227	1.227
KKZC	0.910	0.910	0.910	0.910	0.910	0.910	0.910	0.910	0.910	0.910	0.910	0.910	0.910	0.910	0.910	0.910
KKZZ	0.916	0.916	0.916	0.916	0.916	0.916	0.916	0.916	0.916	0.916	0.916	0.916	0.916	0.916	0.916	0.916
KEBB	0.950	0.950	0.950	0.950	0.950	0.950	0.950	0.950	0.950	0.950	0.950	0.950	0.950	0.950	0.950	0.950
KEBC	0.919	0.919	0.919	0.919	0.919	0.919	0.919	0.919	0.919	0.919	0.919	0.919	0.919	0.919	0.919	0.919
KEBZ	0.968	0.968	0.968	0.968	0.968	0.968	0.968	0.968	0.968	0.968	0.968	0.968	0.968	0.968	0.968	0.968
KECB	0.966	0.966	0.966	0.966	0.966	0.966	0.966	0.966	0.966	0.966	0.966	0.966	0.966	0.966	0.966	0.966
KECC	0.921	0.921	0.921	0.921	0.921	0.921	0.921	0.921	0.921	0.921	0.921	0.921	0.921	0.921	0.921	0.921
KECZ	0.917	0.917	0.917	0.917	0.917	0.917	0.917	0.917	0.917	0.917	0.917	0.917	0.917	0.917	0.917	0.917
KEZB	0.964	0.964	0.964	0.964	0.964	0.964	0.964	0.964	0.964	0.964	0.964	0.964	0.964	0.964	0.964	0.964
KEZC	0.922	0.922	0.922	0.922	0.922	0.922	0.922	0.922	0.922	0.922	0.922	0.922	0.922	0.922	0.922	0.922
KEZZ	0.944	0.944	0.944	0.944	0.944	0.944	0.944	0.944	0.944	0.944	0.944	0.944	0.944	0.944	0.944	0.944
KG5.05BB	0.983	0.983	0.983	0.983	0.983	0.983	0.983	0.983	0.983	0.983	0.983	0.983	0.983	0.983	0.983	0.983
KG5.05BC	0.926	0.926	0.926	0.926	0.926	0.926	0.926	0.926	0.926	0.926	0.926	0.926	0.926	0.926	0.926	0.926
KG5.2BB	0.984	0.984	0.984	0.984	0.984	0.984	0.984	0.984	0.984	0.984	0.984	0.984	0.984	0.984	0.984	0.984
KG5.2BC	0.927	0.927	0.927	0.927	0.927	0.927	0.927	0.927	0.927	0.927	0.927	0.927	0.927	0.927	0.927	0.927
KG5.5BB	0.985	0.985	0.985	0.985	0.985	0.985	0.985	0.985	0.985	0.985	0.985	0.985	0.985	0.985	0.985	0.985
KG5.5BC	0.927	0.927	0.927	0.927	0.927	0.927	0.927	0.927	0.927	0.927	0.927	0.927	0.927	0.927	0.927	0.927
KG5.8BB	0.985	0.985	0.985	0.985	0.985	0.985	0.985	0.985	0.985	0.985	0.985	0.985	0.985	0.985	0.985	0.985
KG5.8BC	0.927	0.927	0.927	0.927	0.927	0.927	0.927	0.927	0.927	0.927	0.927	0.927	0.927	0.927	0.927	0.927
KG9.05BB	0.995	0.995	0.995	0.995	0.995	0.995	0.995	0.995	0.995	0.995	0.995	0.995	0.995	0.995	0.995	0.995
KG9.05BC	0.934	0.934	0.934	0.934	0.934	0.934	0.934	0.934	0.934	0.934	0.934	0.934	0.934	0.934	0.934	0.934
KG9.2BB	0.995	0.995	0.995	0.995	0.995	0.995	0.995	0.995	0.995	0.995	0.995	0.995	0.995	0.995	0.995	0.995
KG9.2BC	0.934	0.934	0.934	0.934	0.934	0.934	0.934	0.934	0.934	0.934	0.934	0.934	0.934	0.934	0.934	0.934
KG9.5BB	0.995	0.995	0.995	0.995	0.995	0.995	0.995	0.995	0.995	0.995	0.995	0.995	0.995	0.995	0.995	0.995
KG9.5BC	0.934	0.934	0.934	0.934	0.934	0.934	0.934	0.934	0.934	0.934	0.934	0.934	0.934	0.934	0.934	0.934
KG9.8BB	0.995	0.995	0.995	0.995	0.995	0.995	0.995	0.995	0.995	0.995	0.995	0.995	0.995	0.995	0.995	0.995
KG9.8BC	0.934	0.934	0.934	0.934	0.934	0.934	0.934	0.934	0.934	0.934	0.934	0.934	0.934	0.934	0.934	0.934
KPBB	1.306	1.306	1.306	1.306	1.306	1.306	1.306	1.306	1.306	1.306	1.306	1.306	1.306	1.306	1.306	1.306
KPBC	0.912	0.912	0.912	0.912	0.912	0.912	0.912	0.912	0.912	0.912	0.912	0.912	0.912	0.912	0.912	0.912
KPBZ	0.907	0.907	0.907	0.907	0.907	0.907	0.907	0.907	0.907	0.907	0.907	0.907	0.907	0.907	0.907	0.907
SKBB	0.965	0.966	0.967	0.967	0.968	0.968	0.969	0.970	0.970	0.971	0.971	0.971	0.971	0.971	0.971	0.971
SKBC	0.905	0.905	0.905	0.905	0.905	0.905	0.905	0.905	0.905	0.905	0.905	0.905	0.905	0.905	0.905	0.905
SKBZ	0.897	0.897	0.897	0.897	0.897	0.896	0.896	0.896	0.896	0.896	0.896	0.896	0.896	0.896	0.896	0.896
SKCB	1.136	1.150	1.154	1.158	1.162	1.166	1.168	1.175	1.185	1.199	1.209	1.218	1.226	1.227	1.228	1.229
SKCC	0.898	0.899	0.899	0.899	0.899	0.899	0.899	0.899	0.899	0.900	0.901	0.902	0.903	0.903	0.903	0.903
SKCZ	0.893	0.893	0.893	0.893	0.893	0.893	0.893	0.893	0.894	0.894	0.895	0.896	0.897	0.898	0.898	0.898
SKZB	1.141	1.156	1.161	1.164	1.169	1.173	1.175	1.182	1.193	1.207	1.217	1.226	1.234	1.235	1.237	1.238
SKZC	0.900	0.900	0.900	0.900	0.900	0.901	0.901	0.901	0.901	0.902	0.903	0.904	0.905	0.905	0.906	0.906
SKZZ	0.894	0.895	0.895	0.895	0.895	0.895	0.895	0.895	0.896	0.897	0.897	0.899	0.900	0.900	0.900	0.900
SEBB	0.933	0.935	0.936	0.937	0.938	0.939	0.940	0.942	0.944	0.946	0.947	0.948	0.949	0.949	0.949	0.949
SEBC	0.903	0.903	0.902	0.902	0.902	0.902	0.902	0.902	0.902	0.902	0.902	0.902	0.902	0.902	0.902	0.902
SEBZ	0.895	0.895	0.895	0.895	0.895	0.894	0.894	0.895	0.895	0.895	0.895	0.895	0.895	0.895	0.895	0.895
SECB	0.964	0.964	0.964	0.965	0.965	0.965	0.966	0.966	0.967	0.967	0.967	0.967	0.967	0.967	0.967	0.967
SECC	0.904	0.904	0.904	0.904	0.904	0.904	0.904	0.904	0.904	0.904	0.904	0.904	0.904	0.904	0.904	0.904
SECZ	0.899	0.899	0.898	0.898	0.898	0.898	0.898	0.898	0.898	0.898	0.898	0.898	0.898	0.898	0.898	0.898
SEZB	0.966	0.966	0.967	0.967	0.967	0.967	0.968	0.968	0.969	0.969	0.970	0.970	0.970	0.970	0.970	0.970
SEZC	0.907	0.907	0.907	0.907	0.907	0.907	0.907	0.907	0.907	0.907	0.907	0.907	0.907	0.907	0.907	0.907
SEZZ	0.898	0.898	0.897	0.897	0.897	0.897	0.897	0.897	0.897	0.897	0.897	0.898	0.898	0.898	0.898	0.898
SG5.05BB	0.984	0.983	0.983	0.982	0.982	0.982	0.982	0.981	0.981	0.980	0.979	0.978	0.978	0.978	0.978	0.978
SG5.05BC	0.902	0.902	0.901	0.901	0.901	0.900	0.900	0.900	0.900	0.900	0.900	0.899	0.899	0.899	0.899	0.899
SG5.05BZ	0.902	0.902	0.901	0.901	0.901	0.900	0.900	0.900	0.900	0.900	0.900	0.899	0.899	0.899	0.899	0.899
SG5.2BB	0.984	0.983	0.983	0.983	0.983	0.982	0.982	0.982	0.981	0.980	0.979	0.978	0.978	0.978	0.978	0.978
SG5.2BC	0.902	0.902	0.901	0.901	0.901	0.901	0.900	0.900	0.900	0.900	0.900	0.899	0.899	0.899	0.899	0.899
SG5.2BZ	0.902	0.902	0.901	0.901	0.901	0.901	0.900	0.900	0.900	0.900	0.900	0.899	0.899	0.899	0.899	0.899
SG5.5BB	0.984	0.983	0.983	0.983	0.983	0.982	0.982	0.982	0.981	0.980	0.979	0.978	0.978	0.978	0.978	0.978
SG5.5BC	0.902	0.902	0.901	0.901	0.901	0.901	0.900	0.900	0.900	0.900	0.900	0.899	0.899	0.899	0.899	0.899
SG5.5BZ	0.902	0.902	0.901	0.901	0.901	0.901	0.900	0.900	0.900	0.900	0.900	0.899	0.899	0.899	0.899	0.899
SG5.8BB	0.984	0.983	0.983	0.983	0.983	0.982	0.982	0.982	0.981	0.980	0.979	0.978	0.978	0.978	0.978	0.978
SG5.8BC	0.902	0.902	0.901	0.901	0.901	0.901	0.900	0.900	0.900	0.900	0.900	0.899	0.899	0.899	0.899	0.899
SG5.8BZ	0.902	0.902	0.901	0.901	0.901	0.901	0.900	0.900	0.900	0.900	0.900	0.899	0.899	0.899	0.899	0.899
SG9.05BB	0.994	0.992	0.990	0.988	0.987	0.987	0.986	0.985	0.984	0.983	0.982	0.981	0.981	0.980	0.980	0.980
SG9.05BC	0.906	0.905	0.904	0.903	0.903	0.903	0.902	0.902	0.902	0.901	0.901	0.901	0.901	0.901	0.901	0.901
SG9.05BZ	0.906	0.905	0.904	0.903	0.903	0.903	0.902	0.902	0.902	0.901	0.901	0.901	0.901	0.901	0.901	0.901
SG9.2BB	0.994	0.992	0.990	0.988	0.987	0.987	0.986	0.985	0.984	0.983	0.982	0.981	0.981	0.980	0.980	0.980
SG9.2BC	0.906	0.905	0.904	0.903	0.903	0.903	0.902	0.902	0.902	0.901	0.901	0.901	0.901	0.901	0.901	0.901
SG9.2BZ	0.906	0.905	0.904	0.903	0.903	0.903	0.902	0.902	0.902	0.901	0.901	0.901	0.901	0.901	0.901	0.901
SG9.5BB	0.994	0.992	0.990	0.988	0.987	0.987	0.986	0.985	0.984	0.983	0.982	0.981	0.981	0.980	0.980	0.980
SG9.5BC	0.906	0.905	0.904	0.903	0.903	0.903	0.902	0.902	0.902	0.901	0.901	0.901	0.901	0.901	0.901	0.901
SG9.5BZ	0.906	0.905	0.904	0.903	0.903	0.903	0.902	0.902	0.902	0.901	0.901	0.901	0.901	0.901	0.901	0.901
SG9.8BB	0.994	0.992	0.990	0.988	0.987	0.987	0.986	0.985	0.984	0.983	0.982	0.981	0.981	0.980	0.980	0.980
SG9.8BC	0.906	0.905	0.904	0.903	0.903	0.903	0.902	0.902	0.902	0.901	0.901	0.901	0.901	0.901	0.901	0.901
SG9.8BZ	0.906	0.905	0.904	0.903	0.903	0.903	0.902	0.902	0.902	0.901	0.901	0.901	0.901	0.901	0.901	0.901
SPBB	1.197	1.214	1.220	1.224	1.230	1.234	1.237	1.247	1.260	1.277	1.291	1.303	1.311	1.313	1.316	1.316
SPBC	0.903	0.902	0.902	0.902	0.902	0.902	0.902	0.903	0.903	0.904	0.905	0.906	0.907	0.908	0.908	0.908
SPBZ	0.896	0.897	0.896	0.896	0.896	0.896	0.897	0.897	0.898	0.899	0.899	0.901	0.902	0.902	0.902	0.902

Tablica A.9: MAE dla algorytmów najbliższych sąsiadów (wybór k najbliższych) cz. 2

	5	10	15	20	25	30	35	40	45	50	60	70	80	90	100	150
KDkBB	0.640	0.624	0.608	0.588	0.557	0.528	0.505	0.492	0.483	0.481	0.478	0.476	0.476	0.476	0.476	0.476
KDkBC	0.652	0.645	0.638	0.629	0.619	0.612	0.605	0.600	0.593	0.591	0.589	0.589	0.589	0.589	0.589	0.589
KDkBZ	0.655	0.650	0.643	0.636	0.628	0.624	0.618	0.613	0.611	0.608	0.605	0.604	0.604	0.604	0.604	0.604
KKBB	0.575	0.559	0.551	0.545	0.541	0.536	0.534	0.534	0.533	0.533	0.533	0.533	0.533	0.533	0.533	0.533
KKBC	0.658	0.663	0.663	0.664	0.663	0.664	0.664	0.664	0.663	0.663	0.663	0.663	0.663	0.663	0.663	0.663
KKBZ	0.663	0.668	0.667	0.668	0.667	0.668	0.670	0.670	0.669	0.669	0.669	0.669	0.669	0.669	0.669	0.669
KKCB	0.588	0.575	0.566	0.560	0.552	0.543	0.535	0.527	0.517	0.509	0.494	0.490	0.490	0.490	0.489	0.489
KKCC	0.659	0.664	0.664	0.667	0.665	0.666	0.665	0.664	0.664	0.663	0.661	0.661	0.661	0.661	0.661	0.661
KKCZ	0.666	0.673	0.672	0.674	0.675	0.675	0.675	0.674	0.674	0.674	0.671	0.671	0.671	0.671	0.671	0.671
KKZB	0.594	0.580	0.571	0.562	0.553	0.542	0.532	0.522	0.510	0.500	0.483	0.477	0.477	0.476	0.476	0.476
KKZC	0.664	0.670	0.669	0.672	0.670	0.671	0.670	0.669	0.669	0.668	0.666	0.666	0.666	0.666	0.666	0.666
KKZZ	0.672	0.677	0.678	0.680	0.680	0.680	0.679	0.678	0.678	0.677	0.674	0.673	0.673	0.673	0.673	0.673
KEBB	0.567	0.563	0.556	0.555	0.554	0.555	0.554	0.553	0.553	0.553	0.552	0.552	0.552	0.552	0.552	0.552
KEBC	0.628	0.638	0.642	0.645	0.647	0.648	0.648	0.649	0.649	0.649	0.649	0.649	0.649	0.649	0.649	0.649
KEBZ	0.619	0.625	0.629	0.631	0.634	0.635	0.636	0.638	0.638	0.638	0.638	0.638	0.639	0.639	0.639	0.639
KECB	0.535	0.527	0.524	0.523	0.522	0.521	0.520	0.519	0.520	0.520	0.520	0.520	0.520	0.520	0.520	0.520
KECC	0.619	0.629	0.633	0.637	0.637	0.639	0.639	0.639	0.639	0.639	0.640	0.640	0.640	0.640	0.640	0.640
KECZ	0.627	0.637	0.640	0.644	0.645	0.646	0.647	0.648	0.648	0.648	0.649	0.649	0.649	0.649	0.649	0.649
KEZB	0.544	0.536	0.530	0.530	0.528	0.529	0.528	0.528	0.528	0.528	0.527	0.527	0.527	0.527	0.527	0.527
KEZC	0.630	0.637	0.642	0.642	0.642	0.644	0.644	0.645	0.646	0.646	0.646	0.646	0.646	0.646	0.646	0.646
KEZZ	0.631	0.637	0.640	0.642	0.643	0.644	0.645	0.645	0.645	0.645	0.645	0.645	0.645	0.645	0.645	0.645
KG5.05BB	0.488	0.493	0.494	0.495	0.489	0.485	0.482	0.481	0.481	0.482	0.481	0.481	0.481	0.481	0.481	0.481
KG5.05BC	0.636	0.646	0.648	0.652	0.654	0.653	0.654	0.656	0.656	0.656	0.656	0.656	0.656	0.656	0.656	0.656
KG5.2BB	0.487	0.493	0.495	0.495	0.489	0.485	0.482	0.482	0.482	0.482	0.481	0.481	0.481	0.481	0.481	0.481
KG5.2BC	0.636	0.645	0.648	0.652	0.653	0.653	0.654	0.655	0.656	0.655	0.656	0.656	0.656	0.656	0.656	0.656
KG5.5BB	0.487	0.492	0.494	0.495	0.489	0.485	0.482	0.482	0.482	0.482	0.481	0.481	0.481	0.481	0.481	0.481
KG5.5BC	0.636	0.645	0.648	0.651	0.653	0.653	0.654	0.655	0.655	0.655	0.655	0.656	0.656	0.656	0.656	0.656
KG5.8BB	0.487	0.492	0.494	0.495	0.489	0.485	0.482	0.482	0.482	0.482	0.481	0.481	0.481	0.481	0.481	0.481
KG5.8BC	0.636	0.645	0.648	0.651	0.653	0.653	0.654	0.655	0.655	0.655	0.655	0.656	0.656	0.656	0.656	0.656
KG9.05BB	0.484	0.486	0.486	0.488	0.480	0.477	0.474	0.472	0.473	0.473	0.473	0.473	0.473	0.473	0.473	0.473
KG9.05BC	0.633	0.641	0.643	0.644	0.645	0.645	0.647	0.647	0.647	0.647	0.647	0.647	0.647	0.647	0.647	0.647
KG9.2BB	0.483	0.486	0.486	0.488	0.480	0.477	0.474	0.472	0.473	0.473	0.473	0.473	0.473	0.473	0.473	0.473
KG9.2BC	0.633	0.641	0.643	0.644	0.644	0.645	0.647	0.647	0.647	0.647	0.647	0.647	0.647	0.647	0.647	0.647
KG9.5BB	0.483	0.486	0.486	0.488	0.480	0.477	0.474	0.472	0.473	0.473	0.473	0.473	0.473	0.473	0.473	0.473
KG9.5BC	0.633	0.641	0.643	0.644	0.645	0.645	0.647	0.647	0.647	0.647	0.647	0.647	0.647	0.647	0.647	0.647
KG9.8BB	0.483	0.486	0.486	0.488	0.480	0.477	0.474	0.472	0.473	0.473	0.473	0.473	0.473	0.473	0.473	0.473
KG9.8BC	0.633	0.641	0.643	0.644	0.645	0.645	0.647	0.647	0.647	0.647	0.647	0.647	0.647	0.647	0.647	0.647
KPBB	0.601	0.596	0.585	0.574	0.562	0.554	0.543	0.529	0.515	0.505	0.492	0.490	0.489	0.489	0.489	0.489
KPBC	0.651	0.657	0.658	0.659	0.658	0.657	0.655	0.654	0.653	0.653	0.651	0.651	0.651	0.651	0.651	0.651
KPBZ	0.658	0.665	0.665	0.667	0.666	0.667	0.666	0.663	0.663	0.662	0.660	0.660	0.660	0.660	0.660	0.660
SKBB	0.622	0.621	0.618	0.616	0.615	0.613	0.610	0.607	0.605	0.604	0.601	0.600	0.597	0.595	0.593	0.590
SKBC	0.661	0.670	0.671	0.673	0.674	0.677	0.678	0.679	0.679	0.678	0.678	0.678	0.677	0.677	0.676	0.676
SKBZ	0.660	0.668	0.669	0.674	0.674	0.677	0.679	0.679	0.680	0.679	0.678	0.678	0.678	0.677	0.677	0.676
SKCB	0.624	0.616	0.611	0.605	0.601	0.597	0.593	0.588	0.585	0.582	0.579	0.577	0.575	0.572	0.567	0.561
SKCC	0.672	0.676	0.679	0.678	0.680	0.679	0.681	0.682	0.683	0.683	0.683	0.683	0.683	0.682	0.682	0.680
SKCZ	0.671	0.675	0.677	0.677	0.678	0.679	0.681	0.681	0.682	0.682	0.682	0.682	0.682	0.682	0.683	0.680
SKZB	0.628	0.620	0.614	0.608	0.604	0.600	0.596	0.591	0.588	0.585	0.582	0.580	0.578	0.574	0.569	0.564
SKZC	0.674	0.677	0.681	0.680	0.681	0.681	0.683	0.684	0.685	0.684	0.684	0.685	0.685	0.683	0.683	0.682
SKZZ	0.672	0.676	0.679	0.679	0.680	0.680	0.682	0.682	0.683	0.683	0.683	0.683	0.683	0.683	0.684	0.681
SEBB	0.628	0.632	0.633	0.634	0.634	0.634	0.635	0.634	0.632	0.632	0.631	0.629	0.627	0.626	0.624	0.619
SEBC	0.655	0.662	0.667	0.672	0.676	0.676	0.677	0.678	0.678	0.679	0.679	0.680	0.680	0.680	0.680	0.679
SEBZ	0.650	0.661	0.665	0.671	0.674	0.676	0.678	0.679	0.680	0.681	0.680	0.681	0.680	0.681	0.681	0.681
SECB	0.591	0.595	0.593	0.593	0.592	0.591	0.590	0.590	0.590	0.590	0.591	0.590	0.590	0.590	0.589	0.586
SECC	0.638	0.647	0.655	0.658	0.660	0.662	0.663	0.664	0.664	0.664	0.665	0.667	0.667	0.667	0.668	0.668
SECZ	0.639	0.647	0.653	0.658	0.660	0.661	0.661	0.663	0.663	0.664	0.665	0.666	0.665	0.666	0.667	0.667
SEZB	0.587	0.589	0.590	0.590	0.590	0.589	0.589	0.588	0.589	0.589	0.588	0.589	0.590	0.590	0.591	0.590
SEZC	0.644	0.651	0.656	0.661	0.663	0.666	0.666	0.667	0.667	0.667	0.668	0.669	0.669	0.670	0.670	0.671
SEZZ	0.642	0.652	0.657	0.660	0.662	0.663	0.666	0.666	0.666	0.667	0.668	0.668	0.669	0.669	0.669	0.669
SG5.05BB	0.559	0.557	0.557	0.556	0.556	0.558	0.560	0.560	0.559	0.559	0.558	0.558	0.557	0.557	0.558	0.559
SG5.05BC	0.627	0.637	0.640	0.642	0.646	0.652	0.656	0.657	0.659	0.659	0.661	0.662	0.663	0.663	0.663	0.663
SG5.05BZ	0.627	0.637	0.640	0.642	0.646	0.652	0.656	0.657	0.659	0.659	0.661	0.662	0.663	0.663	0.663	0.663
SG5.2BB	0.559	0.557	0.556	0.555	0.556	0.558	0.560	0.559	0.559	0.559	0.558	0.557	0.556	0.557	0.556	0.559
SG5.2BC	0.627	0.637	0.639	0.642	0.646	0.652	0.655	0.655	0.658	0.659	0.661	0.661	0.662	0.662	0.662	0.662
SG5.2BZ	0.627	0.637	0.639	0.642	0.646	0.652	0.655	0.655	0.658	0.659	0.661	0.661	0.662	0.662	0.662	0.662
SG5.5BB	0.559	0.557	0.556	0.555	0.556	0.558	0.560	0.559	0.559	0.559	0.558	0.557	0.556	0.557	0.556	0.558
SG5.5BC	0.628	0.637	0.639	0.642	0.646	0.652	0.655	0.655	0.658	0.659	0.661	0.661	0.662	0.662	0.662	0.662
SG5.5BZ	0.628	0.637	0.639	0.642	0.646	0.652	0.655	0.655	0.658	0.659	0.661	0.661	0.662	0.662	0.662	0.662
SG5.8BB	0.559	0.557	0.556	0.555	0.556	0.558	0.560	0.559	0.559	0.559	0.558	0.557	0.556	0.557	0.556	0.558
SG5.8BC	0.628	0.637	0.639	0.642	0.646	0.652	0.655	0.655	0.658	0.659	0.661	0.661	0.662	0.662	0.662	0.662
SG5.8BZ	0.628	0.637	0.639	0.642	0.646	0.652	0.655	0.655	0.658	0.659	0.661	0.661	0.662	0.662	0.662	0.662
SG9.05BB	0.559	0.558	0.557	0.553	0.552	0.552	0.550	0.549	0.548	0.549	0.548	0.547	0.547	0.548	0.546	0.548
SG9.05BC	0.618	0.630	0.624	0.624	0.633	0.641	0.645	0.646	0.650	0.653	0.652	0.651	0.652	0.653	0.653	0.656
SG9.05BZ	0.618	0.630	0.624	0.624	0.633	0.641	0.645	0.646	0.650	0.653	0.652	0.651	0.652	0.653	0.653	0.656
SG9.2BB	0.559	0.558	0.557	0.553	0.552	0.552	0.550	0.549	0.547	0.549	0.548	0.547	0.546	0.548	0.547	0.548
SG9.2BC	0.618	0.630	0.625	0.624	0.633	0.641	0.645	0.646	0.650	0.653	0.652	0.651	0.652	0.653	0.653	0.656
SG9.2BZ	0.618	0.630	0.625	0.624	0.633	0.641	0.645	0.646	0.650	0.653	0.652	0.651	0.652	0.653	0.653	0.656
SG9.5BB	0.559	0.558	0.557	0.553	0.552	0.552	0.550	0.549	0.547	0.549	0.548	0.547	0.546	0.548	0.547	0.548
SG9.5BC	0.618	0.630	0.625	0.624	0.633	0.641	0.645	0.646	0.650	0.653	0.652	0.651	0.652	0.653	0.653	0.656
SG9.5BZ	0.618	0.630	0.625	0.624	0.633	0.641	0.645	0.646	0.650	0.653	0.652	0.651	0.652	0.653	0.653	0.656
SG9.8BB	0.559	0.558	0.557	0.553	0.552	0.552	0.550	0.549	0.547	0.549	0.548	0.547	0.546	0.548	0.547	0.548
SG9.8BC	0.618	0.630	0.625	0.624	0.633	0.641	0.645	0.646	0.650	0.653	0.652	0.651	0.652	0.653	0.653	0.656
SG9.8BZ	0.618	0.630	0.625	0.624	0.633	0.641	0.645	0.646	0.650	0.653	0.652	0.651	0.652	0.653	0.653	0.656
SPBB	0.598	0.592	0.586	0.583	0.581	0.578	0.575	0.570	0.567	0.567	0.562	0.560	0.557	0.553	0.546	0.540
SPBC	0.655	0.666	0.669	0.668	0.670	0.670	0.670	0.671	0.669	0.670	0.669	0.670	0.670	0.670	0.671	0.670
SPBZ	0.657	0.665	0.666	0.667	0.669	0.669	0.669	0.670	0.670	0.669	0.670	0.670	0.670	0.670	0.669	0.668

Tablica A.10: MSPA dla algorytmów najbliższych sąsiadów (wybór k najbliższych) cz. 1

	200	250	300	350	400	450	500	600	700	800	900	1000	1200	1400	1600	1800
KDkBB	0.476	0.476	0.476	0.476	0.476	0.476	0.476	0.476	0.476	0.476	0.476	0.476	0.476	0.476	0.476	0.476
KDkBC	0.589	0.589	0.589	0.589	0.589	0.589	0.589	0.589	0.589	0.589	0.589	0.589	0.589	0.589	0.589	0.589
KDkBZ	0.604	0.604	0.604	0.604	0.604	0.604	0.604	0.604	0.604	0.604	0.604	0.604	0.604	0.604	0.604	0.604
KKBB	0.533	0.533	0.533	0.533	0.533	0.533	0.533	0.533	0.533	0.533	0.533	0.533	0.533	0.533	0.533	0.533
KKBC	0.663	0.663	0.663	0.663	0.663	0.663	0.663	0.663	0.663	0.663	0.663	0.663	0.663	0.663	0.663	0.663
KKBZ	0.669	0.669	0.669	0.669	0.669	0.669	0.669	0.669	0.669	0.669	0.669	0.669	0.669	0.669	0.669	0.669
KKCB	0.489	0.489	0.489	0.489	0.489	0.489	0.489	0.489	0.489	0.489	0.489	0.489	0.489	0.489	0.489	0.489
KKCC	0.661	0.661	0.661	0.661	0.661	0.661	0.661	0.661	0.661	0.661	0.661	0.661	0.661	0.661	0.661	0.661
KKCZ	0.671	0.671	0.671	0.671	0.671	0.671	0.671	0.671	0.671	0.671	0.671	0.671	0.671	0.671	0.671	0.671
KKZB	0.476	0.476	0.476	0.476	0.476	0.476	0.476	0.476	0.476	0.476	0.476	0.476	0.476	0.476	0.476	0.476
KKZC	0.666	0.666	0.666	0.666	0.666	0.666	0.666	0.666	0.666	0.666	0.666	0.666	0.666	0.666	0.666	0.666
KKZZ	0.673	0.673	0.673	0.673	0.673	0.673	0.673	0.673	0.673	0.673	0.673	0.673	0.673	0.673	0.673	0.673
KEBB	0.552	0.552	0.552	0.552	0.552	0.552	0.552	0.552	0.552	0.552	0.552	0.552	0.552	0.552	0.552	0.552
KEBC	0.649	0.649	0.649	0.649	0.649	0.649	0.649	0.649	0.649	0.649	0.649	0.649	0.649	0.649	0.649	0.649
KEBZ	0.639	0.639	0.639	0.639	0.639	0.639	0.639	0.639	0.639	0.639	0.639	0.639	0.639	0.639	0.639	0.639
KECB	0.520	0.520	0.520	0.520	0.520	0.520	0.520	0.520	0.520	0.520	0.520	0.520	0.520	0.520	0.520	0.520
KECC	0.640	0.640	0.640	0.640	0.640	0.640	0.640	0.640	0.640	0.640	0.640	0.640	0.640	0.640	0.640	0.640
KECZ	0.649	0.649	0.649	0.649	0.649	0.649	0.649	0.649	0.649	0.649	0.649	0.649	0.649	0.649	0.649	0.649
KEZB	0.527	0.527	0.527	0.527	0.527	0.527	0.527	0.527	0.527	0.527	0.527	0.527	0.527	0.527	0.527	0.527
KEZC	0.646	0.646	0.646	0.646	0.646	0.646	0.646	0.646	0.646	0.646	0.646	0.646	0.646	0.646	0.646	0.646
KEZZ	0.645	0.645	0.645	0.645	0.645	0.645	0.645	0.645	0.645	0.645	0.645	0.645	0.645	0.645	0.645	0.645
KG5.05BB	0.481	0.481	0.481	0.481	0.481	0.481	0.481	0.481	0.481	0.481	0.481	0.481	0.481	0.481	0.481	0.481
KG5.05BC	0.656	0.656	0.656	0.656	0.656	0.656	0.656	0.656	0.656	0.656	0.656	0.656	0.656	0.656	0.656	0.656
KG5.2BB	0.481	0.481	0.481	0.481	0.481	0.481	0.481	0.481	0.481	0.481	0.481	0.481	0.481	0.481	0.481	0.481
KG5.2BC	0.656	0.656	0.656	0.656	0.656	0.656	0.656	0.656	0.656	0.656	0.656	0.656	0.656	0.656	0.656	0.656
KG5.5BB	0.481	0.481	0.481	0.481	0.481	0.481	0.481	0.481	0.481	0.481	0.481	0.481	0.481	0.481	0.481	0.481
KG5.5BC	0.656	0.656	0.656	0.656	0.656	0.656	0.656	0.656	0.656	0.656	0.656	0.656	0.656	0.656	0.656	0.656
KG5.8BB	0.481	0.481	0.481	0.481	0.481	0.481	0.481	0.481	0.481	0.481	0.481	0.481	0.481	0.481	0.481	0.481
KG5.8BC	0.656	0.656	0.656	0.656	0.656	0.656	0.656	0.656	0.656	0.656	0.656	0.656	0.656	0.656	0.656	0.656
KG9.05BB	0.473	0.473	0.473	0.473	0.473	0.473	0.473	0.473	0.473	0.473	0.473	0.473	0.473	0.473	0.473	0.473
KG9.05BC	0.647	0.647	0.647	0.647	0.647	0.647	0.647	0.647	0.647	0.647	0.647	0.647	0.647	0.647	0.647	0.647
KG9.2BB	0.473	0.473	0.473	0.473	0.473	0.473	0.473	0.473	0.473	0.473	0.473	0.473	0.473	0.473	0.473	0.473
KG9.2BC	0.647	0.647	0.647	0.647	0.647	0.647	0.647	0.647	0.647	0.647	0.647	0.647	0.647	0.647	0.647	0.647
KG9.5BB	0.473	0.473	0.473	0.473	0.473	0.473	0.473	0.473	0.473	0.473	0.473	0.473	0.473	0.473	0.473	0.473
KG9.5BC	0.647	0.647	0.647	0.647	0.647	0.647	0.647	0.647	0.647	0.647	0.647	0.647	0.647	0.647	0.647	0.647
KG9.8BB	0.473	0.473	0.473	0.473	0.473	0.473	0.473	0.473	0.473	0.473	0.473	0.473	0.473	0.473	0.473	0.473
KG9.8BC	0.647	0.647	0.647	0.647	0.647	0.647	0.647	0.647	0.647	0.647	0.647	0.647	0.647	0.647	0.647	0.647
KPBB	0.489	0.489	0.489	0.489	0.489	0.489	0.489	0.489	0.489	0.489	0.489	0.489	0.489	0.489	0.489	0.489
KPBC	0.651	0.651	0.651	0.651	0.651	0.651	0.651	0.651	0.651	0.651	0.651	0.651	0.651	0.651	0.651	0.651
KPBZ	0.660	0.660	0.660	0.660	0.660	0.660	0.660	0.660	0.660	0.660	0.660	0.660	0.660	0.660	0.660	0.660
SKBB	0.588	0.587	0.586	0.584	0.583	0.583	0.582	0.581	0.581	0.581	0.580	0.580	0.580	0.580	0.580	0.580
SKBC	0.675	0.674	0.674	0.674	0.674	0.674	0.674	0.674	0.674	0.674	0.675	0.675	0.675	0.675	0.675	0.675
SKBZ	0.675	0.675	0.675	0.675	0.675	0.674	0.674	0.675	0.675	0.675	0.675	0.675	0.675	0.675	0.675	0.675
SKCB	0.564	0.564	0.563	0.563	0.561	0.561	0.561	0.563	0.563	0.564	0.570	0.572	0.577	0.577	0.580	0.580
SKCC	0.679	0.678	0.678	0.678	0.678	0.677	0.677	0.676	0.676	0.675	0.674	0.672	0.672	0.671	0.671	0.671
SKCZ	0.679	0.679	0.679	0.679	0.679	0.678	0.678	0.677	0.677	0.676	0.675	0.673	0.673	0.673	0.673	0.673
SKZB	0.566	0.566	0.565	0.565	0.563	0.563	0.563	0.565	0.565	0.566	0.571	0.573	0.578	0.579	0.581	0.582
SKZC	0.680	0.679	0.679	0.679	0.679	0.679	0.678	0.677	0.677	0.676	0.674	0.673	0.672	0.672	0.672	0.672
SKZZ	0.680	0.679	0.679	0.679	0.679	0.679	0.678	0.678	0.677	0.676	0.675	0.673	0.673	0.672	0.672	0.672
SEBB	0.613	0.612	0.610	0.609	0.608	0.607	0.605	0.603	0.601	0.599	0.597	0.596	0.595	0.595	0.595	0.595
SEBC	0.678	0.678	0.677	0.678	0.677	0.677	0.677	0.677	0.677	0.677	0.677	0.676	0.676	0.676	0.676	0.676
SEBZ	0.679	0.678	0.677	0.677	0.677	0.677	0.677	0.677	0.677	0.677	0.677	0.677	0.678	0.677	0.678	0.678
SECB	0.586	0.584	0.584	0.583	0.582	0.581	0.581	0.580	0.579	0.579	0.578	0.578	0.578	0.578	0.578	0.578
SECC	0.667	0.667	0.666	0.667	0.667	0.667	0.667	0.667	0.667	0.667	0.667	0.667	0.666	0.666	0.666	0.666
SECZ	0.666	0.666	0.665	0.665	0.666	0.666	0.666	0.666	0.666	0.666	0.666	0.666	0.666	0.666	0.666	0.666
SEZB	0.589	0.589	0.589	0.588	0.587	0.587	0.587	0.584	0.583	0.582	0.581	0.580	0.580	0.580	0.580	0.580
SEZC	0.671	0.671	0.671	0.671	0.671	0.671	0.671	0.671	0.672	0.672	0.671	0.671	0.672	0.672	0.672	0.672
SEZZ	0.669	0.669	0.669	0.669	0.669	0.669	0.670	0.669	0.669	0.669	0.669	0.669	0.669	0.669	0.669	0.669
SG5.05BB	0.562	0.563	0.564	0.565	0.565	0.566	0.566	0.568	0.569	0.569	0.571	0.571	0.572	0.572	0.572	0.572
SG5.05BC	0.665	0.665	0.666	0.665	0.665	0.665	0.665	0.665	0.665	0.665	0.668	0.668	0.668	0.667	0.667	0.667
SG5.05BZ	0.665	0.665	0.666	0.665	0.665	0.665	0.665	0.665	0.665	0.665	0.668	0.668	0.668	0.667	0.667	0.667
SG5.2BB	0.561	0.563	0.563	0.564	0.565	0.565	0.566	0.567	0.569	0.569	0.570	0.571	0.572	0.572	0.572	0.572
SG5.2BC	0.663	0.665	0.665	0.665	0.664	0.664	0.664	0.664	0.664	0.664	0.667	0.667	0.667	0.666	0.666	0.666
SG5.2BZ	0.663	0.665	0.665	0.665	0.664	0.664	0.664	0.664	0.664	0.664	0.667	0.667	0.667	0.666	0.666	0.666
SG5.5BB	0.561	0.563	0.563	0.564	0.565	0.565	0.566	0.567	0.569	0.569	0.570	0.571	0.572	0.572	0.572	0.572
SG5.5BC	0.663	0.665	0.665	0.665	0.664	0.664	0.664	0.664	0.664	0.664	0.667	0.667	0.667	0.666	0.666	0.666
SG5.5BZ	0.663	0.665	0.665	0.665	0.664	0.664	0.664	0.664	0.664	0.664	0.667	0.667	0.667	0.666	0.666	0.666
SG5.8BB	0.561	0.563	0.563	0.564	0.565	0.565	0.566	0.567	0.569	0.569	0.570	0.571	0.572	0.572	0.572	0.572
SG5.8BC	0.663	0.665	0.665	0.665	0.664	0.664	0.664	0.664	0.664	0.664	0.667	0.667	0.667	0.666	0.666	0.666
SG5.8BZ	0.663	0.665	0.665	0.665	0.664	0.664	0.664	0.664	0.664	0.664	0.667	0.667	0.667	0.666	0.666	0.666
SG9.05BB	0.550	0.552	0.554	0.556	0.558	0.558	0.559	0.561	0.564	0.565	0.567	0.567	0.568	0.568	0.568	0.568
SG9.05BC	0.656	0.655	0.655	0.658	0.658	0.658	0.659	0.660	0.660	0.661	0.662	0.661	0.661	0.661	0.661	0.661
SG9.05BZ	0.656	0.655	0.655	0.658	0.658	0.658	0.659	0.660	0.660	0.661	0.662	0.661	0.661	0.661	0.661	0.661
SG9.2BB	0.550	0.553	0.554	0.556	0.558	0.558	0.559	0.561	0.564	0.565	0.567	0.567	0.568	0.568	0.568	0.568
SG9.2BC	0.656	0.655	0.655	0.658	0.658	0.658	0.659	0.660	0.660	0.661	0.662	0.661	0.661	0.661	0.661	0.661
SG9.2BZ	0.656	0.655	0.655	0.658	0.658	0.658	0.659	0.660	0.660	0.661	0.662	0.661	0.661	0.661	0.661	0.661
SG9.5BB	0.550	0.553	0.554	0.556	0.558	0.558	0.559	0.561	0.564	0.565	0.567	0.567	0.568	0.568	0.568	0.568
SG9.5BC	0.656	0.655	0.655	0.658	0.658	0.658	0.659	0.660	0.660	0.661	0.662	0.661	0.661	0.661	0.661	0.661
SG9.5BZ	0.656	0.655	0.655	0.658	0.658	0.658	0.659	0.660	0.660	0.661	0.662	0.661	0.661	0.661	0.661	0.661
SG9.8BB	0.550	0.553	0.554	0.556	0.558	0.558	0.559	0.561	0.564	0.565	0.567	0.567	0.568	0.568	0.568	0.568
SG9.8BC	0.656	0.655	0.655	0.658	0.658	0.658	0.659	0.660	0.660	0.661	0.662	0.661	0.661	0.661	0.661	0.661
SG9.8BZ	0.656	0.655	0.655	0.658	0.658	0.658	0.659	0.660	0.660	0.661	0.662	0.661	0.661	0.661	0.661	0.661
SPBB	0.543	0.544	0.543	0.541	0.540	0.539	0.540	0.542	0.539	0.541	0.546	0.549	0.555	0.557	0.559	0.559
SPBC	0.669	0.669	0.668	0.668	0.668	0.668	0.668	0.668	0.667	0.666	0.665	0.664	0.662	0.661	0.661	0.661
SPBZ	0.668	0.668	0.667	0.667	0.667	0.668	0.668	0.667	0.667	0.665	0.665	0.663	0.662	0.661	0.661	0.661

Tablica A.11: MSPA dla algorytmów najbliższych sąsiadów (wybór k najbliższych) cz. 2

	5	10	15	20	25	30	35	40	45	50	60	70	80	90	100	150
KDkBB	0.300	0.269	0.257	0.213	0.163	0.123	0.064	0.057	0.057	0.048	0.071	0.067	0.069	0.068	0.068	0.068
KDkBC	0.373	0.380	0.371	0.373	0.369	0.355	0.350	0.355	0.353	0.348	0.342	0.342	0.342	0.342	0.342	0.342
KDkBZ	0.374	0.379	0.377	0.379	0.361	0.361	0.362	0.362	0.361	0.360	0.355	0.357	0.356	0.357	0.357	0.357
KKBB	0.193	0.206	0.197	0.193	0.205	0.203	0.203	0.200	0.196	0.198	0.200	0.200	0.200	0.199	0.199	0.199
KKBC	0.389	0.404	0.403	0.408	0.407	0.408	0.410	0.409	0.410	0.410	0.410	0.410	0.410	0.410	0.410	0.410
KKBZ	0.373	0.393	0.401	0.403	0.412	0.412	0.411	0.414	0.410	0.413	0.414	0.415	0.415	0.415	0.415	0.415
KKCB	0.239	0.241	0.209	0.200	0.186	0.160	0.141	0.113	0.088	0.059	0.027	0.016	0.009	0.009	0.009	0.009
KKCC	0.389	0.407	0.398	0.398	0.396	0.411	0.402	0.405	0.404	0.419	0.417	0.416	0.417	0.417	0.417	0.417
KKCZ	0.383	0.409	0.406	0.411	0.406	0.406	0.413	0.403	0.402	0.418	0.418	0.418	0.417	0.417	0.417	0.417
KKZB	0.236	0.245	0.213	0.206	0.186	0.150	0.138	0.118	0.087	0.062	0.039	0.024	0.016	0.015	0.014	0.014
KKZC	0.403	0.408	0.412	0.402	0.411	0.421	0.417	0.419	0.415	0.416	0.405	0.402	0.402	0.402	0.402	0.402
KKZZ	0.387	0.411	0.418	0.415	0.413	0.414	0.420	0.418	0.417	0.411	0.405	0.398	0.399	0.399	0.399	0.399
KEBB	0.245	0.264	0.270	0.275	0.283	0.280	0.280	0.282	0.281	0.280	0.277	0.277	0.277	0.277	0.277	0.277
KEBC	0.370	0.385	0.394	0.394	0.397	0.401	0.400	0.401	0.402	0.403	0.402	0.402	0.402	0.402	0.402	0.402
KEBZ	0.339	0.368	0.374	0.380	0.383	0.388	0.391	0.393	0.394	0.393	0.394	0.394	0.394	0.394	0.394	0.394
KECB	0.152	0.178	0.174	0.185	0.181	0.180	0.179	0.182	0.183	0.181	0.182	0.182	0.182	0.182	0.182	0.182
KECC	0.363	0.384	0.394	0.396	0.393	0.396	0.399	0.399	0.401	0.402	0.401	0.401	0.401	0.401	0.401	0.401
KECZ	0.365	0.382	0.388	0.393	0.397	0.394	0.397	0.400	0.399	0.401	0.400	0.400	0.400	0.400	0.400	0.400
KEZB	0.157	0.169	0.173	0.168	0.170	0.170	0.165	0.167	0.168	0.168	0.165	0.166	0.166	0.166	0.166	0.166
KEZC	0.366	0.382	0.393	0.395	0.396	0.400	0.397	0.397	0.396	0.398	0.398	0.399	0.399	0.399	0.399	0.399
KEZZ	0.369	0.388	0.395	0.397	0.396	0.396	0.398	0.399	0.399	0.398	0.398	0.397	0.398	0.398	0.398	0.398
KG5.05BB	0.054	0.076	0.077	0.093	0.093	0.083	0.103	0.102	0.105	0.108	0.110	0.111	0.111	0.111	0.111	0.111
KG5.05BC	0.363	0.372	0.379	0.381	0.382	0.382	0.383	0.381	0.382	0.381	0.382	0.381	0.381	0.381	0.381	0.381
KG5.2BB	0.054	0.074	0.069	0.091	0.092	0.082	0.093	0.092	0.097	0.102	0.102	0.102	0.102	0.102	0.102	0.102
KG5.2BC	0.364	0.372	0.378	0.379	0.381	0.382	0.382	0.381	0.381	0.381	0.381	0.381	0.380	0.380	0.380	0.380
KG5.5BB	0.054	0.074	0.066	0.093	0.093	0.083	0.093	0.092	0.097	0.102	0.103	0.102	0.103	0.103	0.103	0.103
KG5.5BC	0.364	0.371	0.378	0.379	0.381	0.382	0.382	0.381	0.382	0.382	0.381	0.381	0.381	0.381	0.381	0.381
KG5.8BB	0.054	0.074	0.065	0.093	0.093	0.083	0.092	0.092	0.097	0.102	0.103	0.102	0.103	0.103	0.103	0.103
KG5.8BC	0.364	0.371	0.378	0.379	0.381	0.382	0.382	0.381	0.382	0.382	0.381	0.381	0.381	0.381	0.381	0.381
KG9.05BB	0.048	0.054	0.057	0.054	0.056	0.057	0.049	0.059	0.064	0.066	0.066	0.066	0.067	0.067	0.067	0.067
KG9.05BC	0.367	0.372	0.377	0.376	0.377	0.378	0.377	0.376	0.376	0.376	0.376	0.376	0.376	0.376	0.376	0.376
KG9.2BB	0.048	0.054	0.056	0.054	0.057	0.057	0.048	0.060	0.064	0.067	0.066	0.067	0.067	0.067	0.067	0.067
KG9.2BC	0.367	0.372	0.377	0.376	0.377	0.378	0.377	0.377	0.376	0.377	0.376	0.376	0.376	0.376	0.376	0.376
KG9.5BB	0.048	0.054	0.056	0.054	0.057	0.057	0.048	0.060	0.064	0.067	0.066	0.067	0.067	0.067	0.067	0.067
KG9.5BC	0.367	0.372	0.377	0.376	0.377	0.378	0.377	0.377	0.376	0.377	0.376	0.376	0.376	0.376	0.376	0.376
KG9.8BB	0.048	0.054	0.056	0.055	0.056	0.057	0.048	0.060	0.064	0.067	0.066	0.067	0.067	0.067	0.067	0.067
KG9.8BC	0.367	0.372	0.377	0.376	0.377	0.378	0.377	0.377	0.376	0.377	0.376	0.376	0.376	0.376	0.376	0.376
KPBB	0.245	0.237	0.209	0.181	0.159	0.130	0.110	0.082	0.048	0.017	-0.014	-0.021	-0.022	-0.022	-0.023	-0.023
KPBC	0.372	0.404	0.404	0.400	0.394	0.409	0.406	0.402	0.401	0.407	0.407	0.407	0.408	0.407	0.407	0.407
KPBZ	0.384	0.399	0.398	0.405	0.402	0.400	0.409	0.406	0.405	0.410	0.412	0.412	0.412	0.412	0.412	0.412
SKBB	0.347	0.366	0.371	0.372	0.378	0.378	0.378	0.376	0.378	0.377	0.373	0.371	0.368	0.370	0.368	0.363
SKBC	0.380	0.410	0.414	0.421	0.420	0.419	0.426	0.423	0.426	0.424	0.419	0.420	0.418	0.417	0.419	0.417
SKBZ	0.389	0.413	0.421	0.420	0.422	0.424	0.424	0.424	0.422	0.422	0.419	0.422	0.420	0.420	0.422	0.421
SKCB	0.346	0.326	0.322	0.333	0.334	0.329	0.323	0.314	0.307	0.301	0.293	0.289	0.278	0.276	0.272	0.261
SKCC	0.397	0.401	0.407	0.398	0.403	0.404	0.404	0.403	0.410	0.407	0.411	0.411	0.409	0.408	0.406	0.404
SKCZ	0.400	0.405	0.410	0.400	0.404	0.406	0.406	0.406	0.413	0.410	0.410	0.412	0.410	0.410	0.408	0.408
SKZB	0.351	0.358	0.328	0.331	0.329	0.320	0.312	0.302	0.290	0.284	0.276	0.280	0.274	0.268	0.264	0.252
SKZC	0.393	0.416	0.402	0.410	0.412	0.409	0.410	0.407	0.402	0.400	0.402	0.395	0.394	0.395	0.395	0.395
SKZZ	0.393	0.416	0.404	0.411	0.413	0.410	0.414	0.413	0.404	0.402	0.401	0.401	0.400	0.399	0.396	0.395
SEBB	0.334	0.351	0.356	0.365	0.369	0.370	0.372	0.377	0.378	0.380	0.385	0.383	0.384	0.388	0.391	0.391
SEBC	0.357	0.380	0.390	0.396	0.396	0.404	0.409	0.414	0.414	0.415	0.419	0.422	0.423	0.423	0.424	0.424
SEBZ	0.350	0.381	0.390	0.395	0.401	0.407	0.411	0.412	0.414	0.419	0.423	0.424	0.426	0.424	0.425	0.424
SECB	0.284	0.303	0.315	0.320	0.320	0.326	0.332	0.335	0.338	0.338	0.341	0.346	0.345	0.347	0.348	0.350
SECC	0.334	0.351	0.362	0.374	0.376	0.380	0.380	0.384	0.386	0.388	0.389	0.392	0.393	0.394	0.396	0.399
SECZ	0.343	0.362	0.371	0.378	0.382	0.385	0.388	0.391	0.395	0.396	0.397	0.400	0.400	0.401	0.403	0.408
SEZB	0.281	0.305	0.313	0.317	0.322	0.324	0.329	0.333	0.332	0.334	0.341	0.344	0.345	0.346	0.347	0.350
SEZC	0.343	0.367	0.371	0.375	0.384	0.389	0.387	0.389	0.391	0.392	0.398	0.396	0.399	0.397	0.400	0.403
SEZZ	0.345	0.369	0.375	0.384	0.388	0.392	0.393	0.394	0.394	0.395	0.398	0.398	0.401	0.403	0.407	0.408
SG5.05BB	0.324	0.341	0.351	0.359	0.363	0.372	0.373	0.372	0.371	0.375	0.374	0.374	0.374	0.374	0.370	0.375
SG5.05BC	0.340	0.368	0.372	0.379	0.382	0.391	0.396	0.396	0.398	0.397	0.391	0.396	0.397	0.397	0.396	0.398
SG5.05BZ	0.340	0.368	0.372	0.379	0.382	0.391	0.396	0.396	0.398	0.397	0.391	0.396	0.397	0.397	0.396	0.398
SG5.2BB	0.323	0.340	0.351	0.361	0.364	0.371	0.372	0.370	0.372	0.372	0.371	0.373	0.373	0.375	0.370	0.373
SG5.2BC	0.340	0.369	0.372	0.376	0.380	0.390	0.397	0.396	0.399	0.395	0.390	0.395	0.395	0.396	0.396	0.397
SG5.2BZ	0.340	0.369	0.372	0.376	0.380	0.390	0.397	0.396	0.399	0.395	0.390	0.395	0.395	0.396	0.396	0.397
SG5.5BB	0.323	0.339	0.351	0.361	0.364	0.371	0.372	0.371	0.372	0.372	0.371	0.374	0.373	0.375	0.370	0.373
SG5.5BC	0.340	0.367	0.372	0.376	0.381	0.390	0.397	0.396	0.399	0.395	0.391	0.395	0.396	0.397	0.396	0.397
SG5.5BZ	0.340	0.367	0.372	0.376	0.381	0.390	0.397	0.396	0.399	0.395	0.391	0.395	0.396	0.397	0.396	0.397
SG5.8BB	0.323	0.339	0.351	0.361	0.364	0.371	0.372	0.370	0.372	0.372	0.371	0.374	0.373	0.375	0.370	0.373
SG5.8BC	0.340	0.367	0.372	0.376	0.381	0.389	0.397	0.396	0.399	0.395	0.391	0.395	0.396	0.397	0.396	0.397
SG5.8BZ	0.340	0.367	0.372	0.376	0.381	0.389	0.397	0.396	0.399	0.395	0.391	0.395	0.396	0.397	0.396	0.397
SG9.05BB	0.316	0.341	0.348	0.351	0.353	0.355	0.360	0.361	0.365	0.361	0.362	0.360	0.359	0.360	0.362	0.369
SG9.05BC	0.332	0.360	0.371	0.376	0.381	0.384	0.389	0.390	0.390	0.391	0.389	0.388	0.388	0.390	0.392	0.392
SG9.05BZ	0.332	0.360	0.371	0.376	0.381	0.384	0.389	0.390	0.390	0.391	0.389	0.388	0.388	0.390	0.392	0.392
SG9.2BB	0.316	0.341	0.348	0.351	0.353	0.355	0.360	0.361	0.365	0.361	0.362	0.360	0.359	0.360	0.362	0.369
SG9.2BC	0.332	0.358	0.371	0.376	0.381	0.384	0.389	0.389	0.390	0.392	0.389	0.389	0.388	0.391	0.392	0.392
SG9.2BZ	0.332	0.358	0.371	0.376	0.381	0.384	0.389	0.389	0.390	0.392	0.389	0.389	0.388	0.391	0.392	0.392
SG9.5BB	0.316	0.341	0.348	0.351	0.353	0.355	0.360	0.361	0.364	0.361	0.362	0.360	0.359	0.360	0.362	0.369
SG9.5BC	0.332	0.358	0.371	0.376	0.381	0.384	0.389	0.389	0.390	0.392	0.389	0.389	0.388	0.391	0.392	0.392
SG9.5BZ	0.332	0.358	0.371	0.376	0.381	0.384	0.389	0.389	0.390	0.392	0.389	0.389	0.388	0.391	0.392	0.392
SG9.8BB	0.316	0.341	0.348	0.351	0.353	0.355	0.360	0.361	0.364	0.361	0.362	0.360	0.359	0.360	0.362	0.369
SG9.8BC	0.332	0.358	0.371	0.376	0.381	0.384	0.389	0.389	0.390	0.392	0.389	0.389	0.388	0.391	0.392	0.392
SG9.8BZ	0.332	0.358	0.371	0.376	0.381	0.384	0.389	0.389	0.390	0.392	0.389	0.389	0.388	0.391	0.392	0.392
SPBB	0.292	0.298	0.289	0.302	0.297	0.293	0.284	0.276	0.269	0.264	0.256	0.247	0.241	0.235	0.222	0.210
SPBC	0.354	0.373	0.372	0.380	0.387	0.389	0.389	0.390	0.394	0.392	0.397	0.396	0.395	0.394	0.392	0.390
SPBZ	0.364	0.375	0.375	0.388	0.388	0.391	0.393	0.393	0.400	0.400	0.401	0.400	0.400	0.398	0.395	0.397

Tablica A.12: SRCC dla algorytmów najbliższych sąsiadów (wybór k najbliższych) cz. 1

	200	250	300	350	400	450	500	600	700	800	900	1000	1200	1400	1600	1800
KDkBB	0.068	0.068	0.068	0.068	0.068	0.068	0.068	0.068	0.068	0.068	0.068	0.068	0.068	0.068	0.068	0.068
KDkBC	0.342	0.342	0.342	0.342	0.342	0.342	0.342	0.342	0.342	0.342	0.342	0.342	0.342	0.342	0.342	0.342
KDkBZ	0.357	0.357	0.357	0.357	0.357	0.357	0.357	0.357	0.357	0.357	0.357	0.357	0.357	0.357	0.357	0.357
KKBB	0.199	0.199	0.199	0.199	0.199	0.199	0.199	0.199	0.199	0.199	0.199	0.199	0.199	0.199	0.199	0.199
KKBC	0.410	0.410	0.410	0.410	0.410	0.410	0.410	0.410	0.410	0.410	0.410	0.410	0.410	0.410	0.410	0.410
KKBZ	0.415	0.415	0.415	0.415	0.415	0.415	0.415	0.415	0.415	0.415	0.415	0.415	0.415	0.415	0.415	0.415
KKCB	0.009	0.009	0.009	0.009	0.009	0.009	0.009	0.009	0.009	0.009	0.009	0.009	0.009	0.009	0.009	0.009
KKCC	0.417	0.417	0.417	0.417	0.417	0.417	0.417	0.417	0.417	0.417	0.417	0.417	0.417	0.417	0.417	0.417
KKCZ	0.417	0.417	0.417	0.417	0.417	0.417	0.417	0.417	0.417	0.417	0.417	0.417	0.417	0.417	0.417	0.417
KKZB	0.014	0.014	0.014	0.014	0.014	0.014	0.014	0.014	0.014	0.014	0.014	0.014	0.014	0.014	0.014	0.014
KKZC	0.402	0.402	0.402	0.402	0.402	0.402	0.402	0.402	0.402	0.402	0.402	0.402	0.402	0.402	0.402	0.402
KKZZ	0.399	0.399	0.399	0.399	0.399	0.399	0.399	0.399	0.399	0.399	0.399	0.399	0.399	0.399	0.399	0.399
KEBB	0.277	0.277	0.277	0.277	0.277	0.277	0.277	0.277	0.277	0.277	0.277	0.277	0.277	0.277	0.277	0.277
KEBC	0.402	0.402	0.402	0.402	0.402	0.402	0.402	0.402	0.402	0.402	0.402	0.402	0.402	0.402	0.402	0.402
KEBZ	0.394	0.394	0.394	0.394	0.394	0.394	0.394	0.394	0.394	0.394	0.394	0.394	0.394	0.394	0.394	0.394
KECB	0.182	0.182	0.182	0.182	0.182	0.182	0.182	0.182	0.182	0.182	0.182	0.182	0.182	0.182	0.182	0.182
KECC	0.401	0.401	0.401	0.401	0.401	0.401	0.401	0.401	0.401	0.401	0.401	0.401	0.401	0.401	0.401	0.401
KECZ	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400
KEZB	0.166	0.166	0.166	0.166	0.166	0.166	0.166	0.166	0.166	0.166	0.166	0.166	0.166	0.166	0.166	0.166
KEZC	0.399	0.399	0.399	0.399	0.399	0.399	0.399	0.399	0.399	0.399	0.399	0.399	0.399	0.399	0.399	0.399
KEZZ	0.398	0.398	0.398	0.398	0.398	0.398	0.398	0.398	0.398	0.398	0.398	0.398	0.398	0.398	0.398	0.398
KG5.05BB	0.111	0.111	0.111	0.111	0.111	0.111	0.111	0.111	0.111	0.111	0.111	0.111	0.111	0.111	0.111	0.111
KG5.05BC	0.381	0.381	0.381	0.381	0.381	0.381	0.381	0.381	0.381	0.381	0.381	0.381	0.381	0.381	0.381	0.381
KG5.2BB	0.102	0.102	0.102	0.102	0.102	0.102	0.102	0.102	0.102	0.102	0.102	0.102	0.102	0.102	0.102	0.102
KG5.2BC	0.380	0.380	0.380	0.380	0.380	0.380	0.380	0.380	0.380	0.380	0.380	0.380	0.380	0.380	0.380	0.380
KG5.5BB	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103
KG5.5BC	0.381	0.381	0.381	0.381	0.381	0.381	0.381	0.381	0.381	0.381	0.381	0.381	0.381	0.381	0.381	0.381
KG5.8BB	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103
KG5.8BC	0.381	0.381	0.381	0.381	0.381	0.381	0.381	0.381	0.381	0.381	0.381	0.381	0.381	0.381	0.381	0.381
KG9.05BB	0.067	0.067	0.067	0.067	0.067	0.067	0.067	0.067	0.067	0.067	0.067	0.067	0.067	0.067	0.067	0.067
KG9.05BC	0.376	0.376	0.376	0.376	0.376	0.376	0.376	0.376	0.376	0.376	0.376	0.376	0.376	0.376	0.376	0.376
KG9.2BB	0.067	0.067	0.067	0.067	0.067	0.067	0.067	0.067	0.067	0.067	0.067	0.067	0.067	0.067	0.067	0.067
KG9.2BC	0.376	0.376	0.376	0.376	0.376	0.376	0.376	0.376	0.376	0.376	0.376	0.376	0.376	0.376	0.376	0.376
KG9.5BB	0.067	0.067	0.067	0.067	0.067	0.067	0.067	0.067	0.067	0.067	0.067	0.067	0.067	0.067	0.067	0.067
KG9.5BC	0.376	0.376	0.376	0.376	0.376	0.376	0.376	0.376	0.376	0.376	0.376	0.376	0.376	0.376	0.376	0.376
KG9.8BB	0.067	0.067	0.067	0.067	0.067	0.067	0.067	0.067	0.067	0.067	0.067	0.067	0.067	0.067	0.067	0.067
KG9.8BC	0.376	0.376	0.376	0.376	0.376	0.376	0.376	0.376	0.376	0.376	0.376	0.376	0.376	0.376	0.376	0.376
KPBB	-0.023	-0.023	-0.023	-0.023	-0.023	-0.023	-0.023	-0.023	-0.023	-0.023	-0.023	-0.023	-0.023	-0.023	-0.023	-0.023
KPBC	0.407	0.407	0.407	0.407	0.407	0.407	0.407	0.407	0.407	0.407	0.407	0.407	0.407	0.407	0.407	0.407
KPBZ	0.412	0.412	0.412	0.412	0.412	0.412	0.412	0.412	0.412	0.412	0.412	0.412	0.412	0.412	0.412	0.412
SKBB	0.364	0.364	0.368	0.367	0.365	0.366	0.368	0.366	0.366	0.367	0.368	0.368	0.368	0.368	0.368	0.368
SKBC	0.421	0.420	0.423	0.425	0.422	0.423	0.422	0.420	0.420	0.422	0.421	0.421	0.421	0.421	0.421	0.421
SKBZ	0.420	0.420	0.422	0.423	0.421	0.421	0.423	0.422	0.421	0.422	0.423	0.422	0.422	0.421	0.421	0.421
SKCB	0.251	0.249	0.248	0.248	0.244	0.242	0.244	0.251	0.248	0.236	0.240	0.251	0.266	0.272	0.275	0.276
SKCC	0.405	0.404	0.404	0.403	0.403	0.402	0.401	0.401	0.400	0.400	0.402	0.400	0.399	0.399	0.400	0.400
SKCZ	0.407	0.408	0.409	0.410	0.408	0.407	0.407	0.406	0.404	0.404	0.401	0.401	0.400	0.400	0.401	0.401
SKZB	0.240	0.239	0.242	0.242	0.240	0.240	0.243	0.247	0.243	0.238	0.254	0.264	0.269	0.273	0.271	0.271
SKZC	0.395	0.394	0.396	0.395	0.395	0.393	0.393	0.394	0.392	0.388	0.396	0.396	0.388	0.386	0.392	0.392
SKZZ	0.396	0.396	0.399	0.397	0.397	0.397	0.396	0.396	0.392	0.390	0.395	0.394	0.386	0.385	0.390	0.390
SEBB	0.391	0.388	0.390	0.392	0.391	0.390	0.390	0.389	0.389	0.387	0.385	0.385	0.386	0.385	0.385	0.385
SEBC	0.425	0.427	0.428	0.428	0.428	0.427	0.428	0.426	0.427	0.425	0.426	0.426	0.426	0.426	0.426	0.426
SEBZ	0.422	0.421	0.421	0.423	0.424	0.423	0.424	0.424	0.424	0.425	0.425	0.424	0.424	0.424	0.424	0.424
SECB	0.351	0.351	0.351	0.351	0.351	0.353	0.353	0.354	0.354	0.354	0.355	0.355	0.354	0.354	0.354	0.354
SECC	0.400	0.404	0.406	0.407	0.408	0.408	0.410	0.411	0.408	0.409	0.409	0.409	0.410	0.410	0.410	0.410
SECZ	0.409	0.411	0.412	0.412	0.414	0.415	0.417	0.417	0.417	0.416	0.417	0.417	0.417	0.417	0.417	0.417
SEZB	0.349	0.349	0.348	0.350	0.350	0.350	0.350	0.350	0.350	0.352	0.351	0.352	0.352	0.352	0.352	0.352
SEZC	0.406	0.407	0.405	0.404	0.405	0.405	0.407	0.407	0.408	0.408	0.409	0.409	0.409	0.409	0.409	0.409
SEZZ	0.411	0.409	0.410	0.411	0.411	0.411	0.411	0.411	0.412	0.412	0.412	0.413	0.413	0.413	0.413	0.413
SG5.05BB	0.374	0.375	0.378	0.380	0.383	0.385	0.385	0.386	0.383	0.388	0.388	0.389	0.389	0.390	0.390	0.390
SG5.05BC	0.400	0.401	0.403	0.405	0.404	0.402	0.404	0.404	0.404	0.403	0.404	0.404	0.404	0.405	0.405	0.405
SG5.05BZ	0.400	0.401	0.403	0.405	0.404	0.402	0.404	0.404	0.404	0.403	0.404	0.404	0.404	0.405	0.405	0.405
SG5.2BB	0.373	0.376	0.378	0.380	0.383	0.385	0.385	0.386	0.382	0.388	0.387	0.388	0.389	0.389	0.390	0.390
SG5.2BC	0.399	0.400	0.403	0.404	0.403	0.403	0.403	0.404	0.404	0.403	0.403	0.404	0.404	0.404	0.404	0.404
SG5.2BZ	0.399	0.400	0.403	0.404	0.403	0.403	0.403	0.404	0.404	0.403	0.403	0.404	0.404	0.404	0.404	0.404
SG5.5BB	0.373	0.376	0.379	0.380	0.383	0.385	0.385	0.386	0.383	0.388	0.387	0.388	0.389	0.389	0.390	0.390
SG5.5BC	0.399	0.400	0.403	0.404	0.403	0.403	0.403	0.404	0.404	0.403	0.404	0.404	0.404	0.404	0.404	0.404
SG5.5BZ	0.399	0.400	0.403	0.404	0.403	0.403	0.403	0.404	0.404	0.403	0.404	0.404	0.404	0.404	0.404	0.404
SG5.8BB	0.373	0.376	0.379	0.380	0.383	0.385	0.385	0.386	0.382	0.388	0.387	0.388	0.389	0.389	0.390	0.390
SG5.8BC	0.399	0.400	0.403	0.404	0.403	0.403	0.403	0.404	0.404	0.403	0.404	0.404	0.404	0.404	0.404	0.404
SG5.8BZ	0.399	0.400	0.403	0.404	0.403	0.403	0.403	0.404	0.404	0.403	0.404	0.404	0.404	0.404	0.404	0.404
SG9.05BB	0.368	0.371	0.378	0.378	0.378	0.382	0.383	0.384	0.385	0.386	0.387	0.388	0.389	0.390	0.390	0.390
SG9.05BC	0.395	0.396	0.400	0.400	0.400	0.401	0.401	0.402	0.402	0.402	0.402	0.403	0.404	0.404	0.403	0.403
SG9.05BZ	0.395	0.396	0.400	0.400	0.400	0.401	0.401	0.402	0.402	0.402	0.402	0.403	0.404	0.404	0.403	0.403
SG9.2BB	0.367	0.371	0.378	0.378	0.378	0.382	0.383	0.384	0.385	0.386	0.387	0.388	0.389	0.390	0.390	0.390
SG9.2BC	0.395	0.396	0.400	0.400	0.400	0.401	0.401	0.402	0.402	0.402	0.402	0.403	0.404	0.404	0.404	0.404
SG9.2BZ	0.395	0.396	0.400	0.400	0.400	0.401	0.401	0.402	0.402	0.402	0.402	0.403	0.404	0.404	0.404	0.404
SG9.5BB	0.367	0.371	0.378	0.378	0.378	0.382	0.383	0.384	0.385	0.386	0.387	0.388	0.389	0.390	0.390	0.390
SG9.5BC	0.395	0.396	0.400	0.400	0.400	0.401	0.401	0.402	0.402	0.402	0.402	0.403	0.404	0.404	0.404	0.404
SG9.5BZ	0.395	0.396	0.400	0.400	0.400	0.401	0.401	0.402	0.402	0.402	0.402	0.403	0.404	0.404	0.404	0.404
SG9.8BB	0.367	0.371	0.378	0.378	0.378	0.382	0.383	0.384	0.385	0.386	0.387	0.388	0.389	0.390	0.390	0.390
SG9.8BC	0.395	0.396	0.400	0.400	0.400	0.401	0.401	0.402	0.402	0.402	0.402	0.403	0.404	0.404	0.404	0.404
SG9.8BZ	0.395	0.396	0.400	0.400	0.400	0.401	0.401	0.402	0.402	0.402	0.402	0.403	0.404	0.404	0.404	0.404
SPBB	0.197	0.195	0.198	0.201	0.193	0.193	0.194	0.198	0.197	0.188	0.192	0.203	0.221	0.225	0.229	0.228
SPBC	0.391	0.391	0.392	0.393	0.391	0.391	0.391	0.389	0.389	0.389	0.389	0.383	0.382	0.382	0.382	0.382
SPBZ	0.394	0.395	0.396	0.397	0.397	0.397	0.397	0.396	0.394	0.393	0.392	0.389	0.387	0.387	0.387	0.387

Tablica A.13: SRCC dla algorytmów najbliższych sąsiadów (wybór k najbliższych) cz. 2

typ	regularyzacja	f	RMSE	MAE	MSPA	SRCC
bazowy	0.001	–	1.0844	0.9056	0.6738	0.387
bazowy	0.01	–	1.0852	0.9061	0.6731	0.3864
bazowy	0.1	–	1.0805	0.9049	0.6731	0.3852
bazowy	1	–	1.0712	0.9082	0.66	0.3902
svd	0.001	10	1.1162	0.9193	0.6555	0.3631
svd	0.001	20	1.1154	0.9203	0.6543	0.3573
svd	0.001	40	1.1207	0.9227	0.6472	0.3608
svd	0.01	10	1.1181	0.9206	0.6517	0.3648
svd	0.01	20	1.1167	0.9208	0.6501	0.3666
svd	0.01	40	1.1164	0.921	0.6486	0.3638
svd	0.1	10	1.1091	0.9175	0.6555	0.3664
svd	0.1	20	1.0884	0.9132	0.651	0.3808
svd	0.1	40	1.0989	0.9167	0.6478	0.3735
svd	1	10	1.0726	0.9095	0.6593	0.3896
svd	1	20	1.075	0.912	0.6556	0.3887
svd	1	40	1.0739	0.911	0.6552	0.3892

Tablica A.14: Wyniki dla modelowania

Dodatek B

Zawartość płyt DVD

Do pracy została załączona płyta DVD, która zawiera:

- kod źródłowy niniejszej pracy w formacie LaTeX;
- niniejszą pracę w formacie pdf;
- skrypty w języku R użyte podczas analiz przeprowadzonych na potrzeby niniejszej pracy;
- wyniki analiz przeprowadzonych na potrzeby niniejszej pracy.

Bibliografia

[1] Agarwal D., Chen B.C., *Machine Learning for Large Scale Recommender Systems.* Tutorial presented at 28th International Conference on Machine Learning, 2011.

[2] Bell R.M., Koren Y., *Scalable Collaborative Filtering with Jointly Derived Neighborhood Interpolation Weights.* Proceedings of the 7th IEEE International Conference on Data Mining, 2007.

[3] Bell R., Koren Y., Volinsky C., *Modeling relationships at multiple scales to improve accuracy of large recommender systems.* Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2007

[4] Blei D., Ng A., Jordan M., *Latent Dirichlet Allocation.* Journal of Machine Learning Research, 2003.

[5] Breese J., Heckerman D., Kadie C., *Empirical Analysis of Predictive Algorithms for Collaborative Filtering.* Proceedings of the 14th Conference on Uncertainty in Artifical Intelligence, 1998.

[6] Deerwester S., Dumais S.T., Furnas G.W., Landauer T.K., Harshman R., *Indexing by Latent Semantic Analysis.* Journal of the Society for Information Science 41, 1990.

[7] Fouss F., Pirotte A., Renders J.M., Saerens M., *Random-Walk Computation of Similarities between Nodes of a Graph with Application to Collaborative Recommendation.* IEEE Transactions on Knowledge and Data Engineering 19, 2007.

[8] Funk S., *Netflix Update: Try This At Home.* `http://sifter.org/~simon/journal/20061211.html`, 2006.

[9] Giori M., Pucci A., *ItemRank: A Random-Walk Based Scoring Algorithm for Recommender Engines.* Proceedings of the 2007 IJCAI Conference, 2007.

[10] Herlocker J.L., Konstan J.A., Borchers A., Riedl J. *An algorithmic framework for performing collaborative filtering.* Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 1999

[11] Hofmann T., *Latent Semantic Models for Collaborative Filtering.* ACM Transactions on Information Systems 22, 2004.

[12] Huang Z., Chen H., Zeng D., *Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering.* ACM Transactions on Information Systems 22, 2004.

[13] Jannach D., Friedrich G., *Tutorial: Recommender Systems.* Tutorial presented at International Joint Conference on Artificial Intelligence, 2011.

[14] Jannach D., Zanker M., Felfernig A., Friedrich G., *Recommender Systems. An Introduction.* Cambridge University Press, 2011.

[15] Kim D., Yum B.,*Collaborative Filtering Based on Iterative Principal Component Analysis.* Expert Systems with Applications 28, 2005.

[16] Konstan J., Riedl J., *AI Techniques for Personalized Recommendation.* Tutorial presented at International Joint Conference on Artificial Intelligence, 2003.

[17] Koren Y., *Factorization Meets the Neighborhood: a Multifaceted Collaborative Filtering Model.* Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2008.

[18] Koren Y., *The BellKor Solution to the Netflix Grand Prize.*, 2009.

[19] Lathia N., Hailes S., Capra L., *The Effect of Correlation Coefficients on Communities of Recommenders.* Proceedings of the 2008 ACM symposium on Applied computing, 2008.

[20] Lemire D., Maclachan A., *Slope One Predictors for Online Rating-Based Collaborative Filtering.* SIAM Data Mining, 2005.

[21] Luo H., Niu C., Shen R., Ullrich C., *A collaborative filtering framework based on both local user similarity and global user similarity.* Machine Learning 72, 2008.

[22] Ma H., King I., Lyu M.R., *Effective missing data prediction for collaborative filtering.* Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, 2007

[23] Marlin B., Zemel R., *The multiple multiplicative factor model for collaborative filtering.* ACM International Conferences Proceeding Series 69, 2004.

[24] *Netflix.* `http://www.netflix.com/`

[25] *Netflix Prize.* `http://www.netflixprize.com/`

[26] Paterek A., *Improving Regularized Singular Value Decomposition for Collaborative Filtering.* Proceedings of the KDD Cup and Workshop, 2007.

[27] Piotte M., Chabbert M., *The Pragmatic Theory Solution to the Netflix Grand Prize.*, 2009.

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

[29] Sarwar B.M., Karypis G., Konstan J.A., Riedl J., *Application of Dimensionality Reduction in Recommender System – A Case Study.* ACM WebKDD Workshop, 2000.

[30] Shani G., Brafman R., Heckerman D., *An MDP-Based Recommender System.* Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence, 2002.

[31] Spertus E., Sahami M., Buyukkoten O., *Evaluating Similarity Measures: A Large-Scale Study in the Orkut Social Network.* Proceedings of the 11th ACM SIGKDD international conference on Knowledge discovery in data mining, 2005.

[32] Takacs G., Pilaszy I., Nemeth B., Tikk D., *Matrix Factorization and Neighbor Based Algorithms for the Netflix Prize Problem.* Proceedings of the 2008 ACM conference on Recommender systems, 2008.

[33] Takacs G., Pilaszy I., Nemeth B., Tikk D., *Scalable Collaborative Filtering Approaches for Large Recommender Systems.* Journal of Machine Learning Research 10, 2009.

[34] Toscher A., Jahrer M., *The BigChaos Solution to the Netflix Grand Prize.*, 2009.

[35] Zhang T., *Solving Large Scale Linear Prediction Problems Using Stochastic Gradient Descent Algorithms.* Proceedings of the 21st International Conference on Machine Learning , 2004.