|  |  |  |
| --- | --- | --- |
|  | **Софийски университет „Св. Кл. Охридски”**  Факултет по математика и информатика |  |
|  |  |  |

**Курсов Проект**

на тема:

„ Използване на Wikipedia за подобряване на SVD базирана препоръчваща система”

Студент: **Калоян Спиридонов, Денис Михайлов**

Факултетен номер: **25750, 25788**

Курс: I

Преподавател: **д-р Милен Чечев**

Дата: София, 29.06.2018г.

Съдържание

[1. Проблем 2](#_Toc517968761)

[2. Съществуващи решения 2](#_Toc517968762)

[3. Избрано решение 4](#_Toc517968763)

[4. Експерименти 4](#_Toc517968764)

[4.1 Как да намерим Wikipedia страницата на даден филм? 4](#_Toc517968765)

[4.2 Изчисляване на близостта между всяка двойка филми. 4](#_Toc517968766)

[4.3 Генериране на изкуствени рейтинги 5](#_Toc517968767)

[4.4 Адаптиране на SVD към сходствата от Wikipedia 5](#_Toc517968768)

[4.4.1 Адаптиране на степента на обучение спрямо данните 5](#_Toc517968769)

[4.4.2 Обучаване на два различни SVD модела 5](#_Toc517968770)

[5. Резултати 6](#_Toc517968771)

[5.1 Адаптиране на степента на обучение спрямо данните 6](#_Toc517968772)

[5.2 Обучаване на два различни SVD модела 6](#_Toc517968773)

[6. Източници 7](#_Toc517968774)

# Проблем

Важна част от препоръчващите системи е предсказването на user-item рейтинги. Предсказването обаче има 2 основни проблема – липса на данни и проблемът със студеното стартиране. В повечето случаи user-rating матрицата съдържа много малко рейтинги. Поради това е трудно да се обучат алгоритми, основаващи се единствено на тези оценки. Проблемът със студеното стартиране е подобен, но се фокусира върху нови елементи, където оценките са трудни за предвиждане, преди да са били налице достатъчно данни.

# Съществуващи решения

Идеята за интегриране на външни източници за подобряване на препоръчващите системи са изучени в предишни проучвания, като се различават по източниците, които използват и по използвания метод за добавяне на знанието към модела. Например [2] използва социални мрежи, за да сметне сходството между потребителите. Възможни индикатори за сходство включват еднакви събития, които потребителите посещават или да са членове на един и същи екип. Сходствата се използват, за да се създадат профили на потребителите, които после да се интегрират в препоръчваща система. Освен това може да се използва съдържанието на външни източници като съдържанието може да включва жанр, актьор, режисьор на филм, за да се изчисли близостта между филмите. Например [3] използва информация от няколко източника на знание като Wikipedia.

# Избрано решение

В този проект ние използваме Wikipedia базирани сходства между елементи в контекста на SVD алгоритъма. Показваме два метода за интегрирането на тези данни - адаптиране на степента на обучение и смес от два различни SVD. Оценяваме ефективността на предложените методи върху MovieLens данните.

# Експерименти

Първата стъпка e намиране на близостта между филмите от MovieLens използвайки информация от Wikipedia. За целта за всеки филм трябва да намерим съответната страница в Wikipedia и да изчислим item-item сходства, чрез които да генерираме изкуствени рейтинги.

## Как да намерим Wikipedia страницата на даден филм?

За целта използваме прост алгоритъм:

* За всеки филм:
  + Вземаме името на филма и генерираме различни негови варианти – премахваме годината, “, The”, “, А“, “, An“, добавяме “(film)”, „(<година> film)” и т.н.
  + За всеки вариант на име:
    - Търсим съответна страница във Wikipedia.
    - Вземаме категориите от страницата и броим в колко от тях се среща думата „film”
    - За оригинална страница избираме тази с най-много срещания на “film”

Този прост алгоритъм успява да намери страниците на 85.5% от филмите от MovieLens-100k dataset .

## Изчисляване на близостта между всяка двойка филми.

След като вече разполагаме с Wikipedia страниците, трябва да намерим тяхната близост.

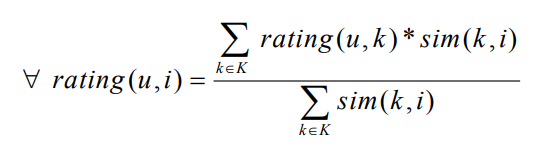
За целта използваме следния алгоритъм:

* За всяка двойка филми (x, y):
* Вземаме wiki страницата на х – page\_x
* Вземаме wiki страницата на y – page\_y
* Извличаме категориите от page\_x – cat\_x
* Извличаме категориите от page\_y – cat\_y
* Пресмятаме jaccard\_similarity(cat\_x, cat\_y)

## Генериране на изкуствени рейтинги

Изкуствените рейтинги които ще използваме по-долу се генерират по следния начин:

За всеки липсващ потребителски рейтинг използваме всички филми които са били оценени от потребителя и Wikipedia близостта. Изчисленията стават по следната формула:



Този метод не слага рейтинги на всички непознати предмети защото много от предметите не са близки с предмети оценени от потребителя. Още повече някои филми нямат асоциирани Wikipedia страници.

## 4.4 Адаптиране на SVD към сходствата от Wikipedia

4.4.1 Адаптиране на степента на обучение спрямо данните

Използваме user-item рейтингите и изкуствените рейтинги за да тренираме SVD модел. Броят на изкуствените рейтинги е много по-голям от броят на потребителските и затова моделът е по-пристрастен към изкуствените. За да премахнем този ефект използваме различни learning rate-и по време на обучението спрямо типа данни. За да определим LR преброяваме броят на оригиналните и изкуствените рейтинги. Намираме пропорцията оригинални/изкуствени и спрямо нея задаваме подходящи стойности.

### 4.4.2 Обучаване на два различни SVD модела

В този подход тренираме два различни SVD модела – един върху оригиналните рейтинги и един върху изкуствените. След това във фазата на предсказване използваме средното от 2-та модела за да определим крайните резултати.

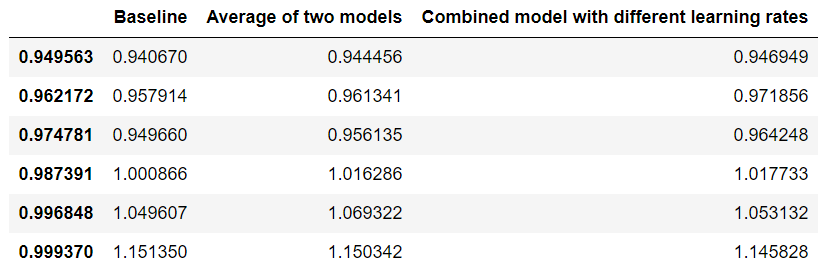
# Резултати

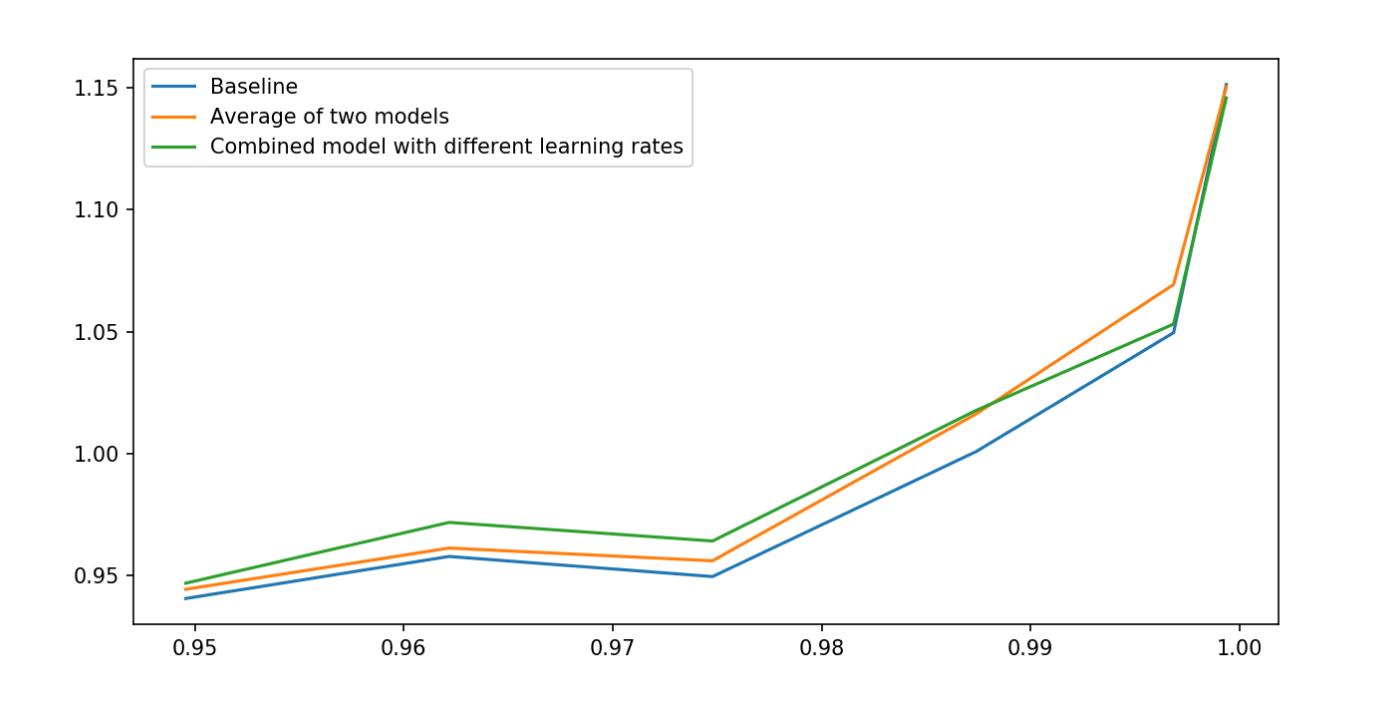
5.1 Адаптиране на степента на обучение спрямо данните

След прилагане на този метод не успяхме да подобрим базовия SVD модел. Тествахме метода на няколко стойности за липсващи данни от 99.95% до 95%. Изглежда, че когато липсата на данни е много голяма наистина леко се подобрява модела, но разликата не изглежда толкова голяма

### 5.2 Обучаване на два различни SVD модела

И този подход не доведе до подобряване на SVD модела и като цяло беше по-слаб от предния модел





# Източници

1. Gilad Katz, Guy Shani, Bracha Shapira, Lior Rokach, Using Wikipedia to Boost SVD Recommender Systems, 2012
2. Middleton, S.E., H. Alani, and D.D. Roure, Exploiting Synergy Between Ontologies and Recommender Systems. CoRR, 2002. cs.LG/0204012.
3. Semeraro, G., Lops P., Basile P., de Gemmis M.,, Knowledge infusion into content-based recommender systems, in Proceedings of the third ACM conference on Recommender systems. 2009, ACM: New York, New York, USA. p. 301- 304.