|  |  |  |
| --- | --- | --- |
|  | **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](#_Toc517959014)

[2. Избрано решение 2](#_Toc517959015)

[3. Експерименти 2](#_Toc517959016)

[3.1 Как да намерим Wikipedia страницата на даден филм? 2](#_Toc517959017)

[3.2 Изчисляване на близостта между всяка двойка филми. 3](#_Toc517959018)

[3.3 Генериране на изкуствени рейтинги 3](#_Toc517959019)

[3.4 Адаптиране на SVD към близостите от Wikipedia 3](#_Toc517959020)

[3.4.1 Адаптиране на степента на обучение спрямо данните 3](#_Toc517959021)

[3.4.2 Обучаване на два различни SVD модела 4](#_Toc517959022)

# 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.

# 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 близости, 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 .тн.
  + 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.

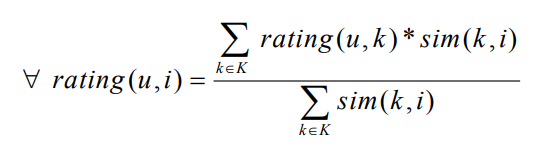
За целта използваме следния алгоритъм:

* 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
* Пресмянтаме 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.

## 3.4 Адаптиране на SVD към близостите от Wikipedia

3.4.1 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.

### 3.4.2 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.