Изготвили:

Сияна славова, 24963

Иван Капукаранов, 24958

Проект за курса „Препоръчващи системи“

Задача: recSys 2013 challenge

Съдържание

[Проблем 2](#_Toc454362794)

[Съществуващи решения 2](#_Toc454362795)

[Избрано решение 2](#_Toc454362796)

[Експерименти 4](#_Toc454362797)

[Параметри 4](#_Toc454362798)

[Резултати 4](#_Toc454362799)

[Графики 4](#_Toc454362800)

[RMSE 6](#_Toc454362801)

[Бъдещо развитие 6](#_Toc454362802)

[Използвана литература 6](#_Toc454362803)

# Проблем

Данните за задачата са предоставени от [Yelp](http://www.yelp.com/), като съдържат над:

* 10,000 фирми (business)
* 8,000 check – in sites
* 40,000 потребителя (users)
* 200,000 ревюта (reviews)

Трябва да се предскаже рейтинга, който потребител ***u*** би дал на продукт ***i***. За целта са дадени обучаващо и тестово множество, като тестовото множество съдържа само колони потребител и фирма.  
  
Оценяването на алгоритъма става чрез използването на средно квадратична грешка.

# Съществуващи решения

Решението на спечелилите състезанието (*BrickMover Team*) е следното:

**Използвани модели:** Matrix Factorization, Linear Regression, Regression Tree, Global Effects. Според това дали има данни за потребителя или фирмата, тест данните се разделят на 4 групи за оценка и моделите са оптимизирани спрямо тях.

В решението са използвани различни features за потребител и фирма, като са генерирани и cross - joined features . Например, ако има две фирми от една и съща категория, но в различни градове, техните оценки могат да зависят от разликата в оценката между градовете. За избиране на features обучаващото множество се разделя на 7 равни части, за да формира локален cross - validation сет.

Постигната грешка: 1.21251

# Избрано решение

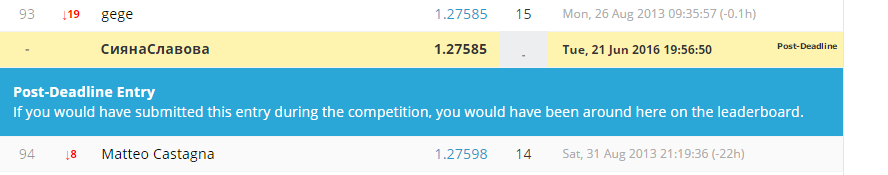
За реализиране на задачата използваме [Apache Mahout](http://mahout.apache.org/). За предварителна обработка на данните и оценяване на алгоритмите използваме [R](https://www.r-project.org/).  
  
След проведените експерименти, описани по – долу в документа, получихме най – малка грешка върху избрани от нас тестови данни – 1.0473 при използване на user – based подхода с neighborhood threshold 0.5. Ако алгоритъма не намери близки потребители спрямо този праг, за оценка се връща средната оценка от средното за потребителя и средното за фирмата, т.е формулата е следната:

(userAverage + businessAverage)/2

Ако няма данни за средното за потребител или business взимаме средното съответно за всички потребители или фирми. (cold start decision)  
Събмитнахме получените оценките върху тестовите данни от състезанието и получихме грешка 1.27585 върху тях.

Постигната грешка върху тестовите данни от състезанието:

Предварителна обработка  
Като предварителна обработка опростихме данните, като взехме само колоните: user\_id, business\_id и stars от обучаващото множество. Тъй като user\_id и business\_id в оригиналния файл са текстови полета, а за да изградим модел на данните трябва id полето да е от числов тип, мапнахме user\_id към ново user\_id. Същото правим и за business\_id.



Пример:

"iUnAEpltJi0MCjmWrPu9w",43872

Преди:  
{"votes": {"funny": 0, "useful": 5, "cool": 2}, "user\_id": "rLtl8ZkDX5vH5nAx9C3q5Q", "review\_id": "fWKvX83p0-ka4JS3dc6E5A", "stars": 5, "date": "2011-01-26", "text": "My wife took me here on my birthday for breakfast and it was excellent. The weather was perfect which made sitting outside overlooking their grounds an absolute pleasure. Our waitress was excellent and our food arrived quickly on the semi-busy Saturday morning. It looked like the place fills up pretty quickly so the earlier you get here the better.\n\nDo yourself a favor and get their Bloody Mary. It was phenomenal and simply the best I've ever had. I'm pretty sure they only use ingredients from their garden and blend them fresh when you order it. It was amazing.\n\nWhile EVERYTHING on the menu looks excellent, I had the white truffle scrambled eggs vegetable skillet and it was tasty and delicious. It came with 2 pieces of their griddled bread with was amazing and it absolutely made the meal complete. It was the best \"toast\" I've ever had.\n\nAnyway, I can't wait to go back!", "type": "review", "business\_id": "9yKzy9PApeiPPOUJEtnvkg"}

След:

24539,7638,5

Тъй като някой потребители са забранили използването на тяхна информация за публични цели, ги няма в множеството на потребителите, но все още може да има останали оценки, оставени от тях, в обучаващото множество. Данните за тези потребители не присъстват в новото обучаващо множество.

За целите на проекта и за да може да измерим RMSE върху тестов сет, ние разделихме оригиналното обучаващо множество на две – тестово и обучаващо, като взехме 30 % случайни записи за тестовото и 70 % - за обучаващото.

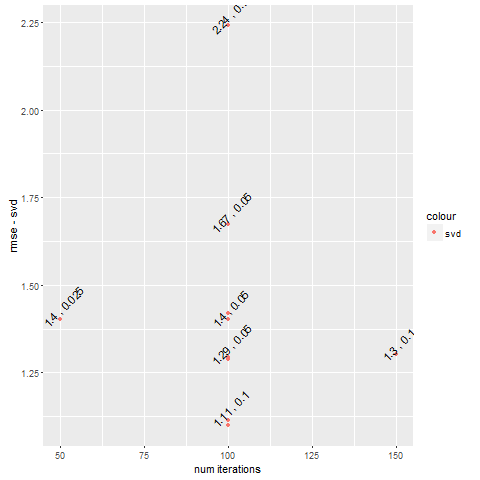
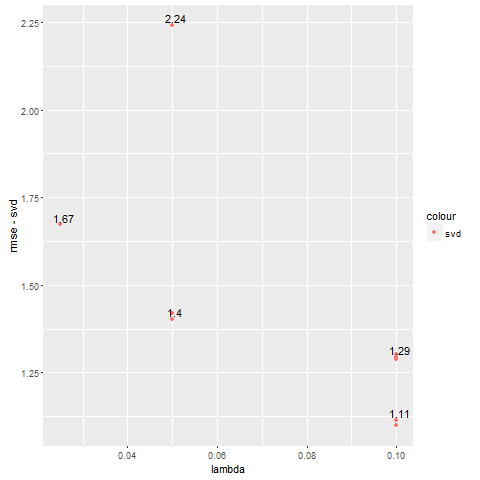
# Експерименти

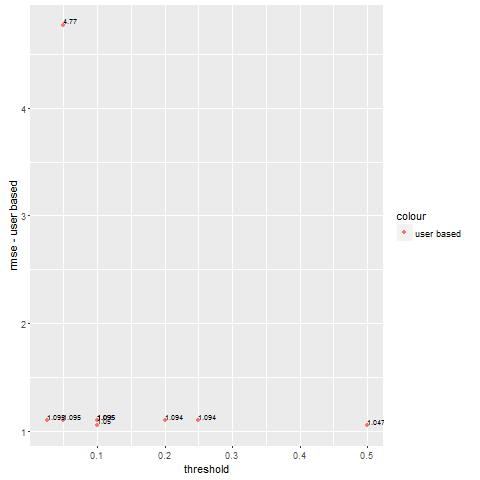
## Параметри

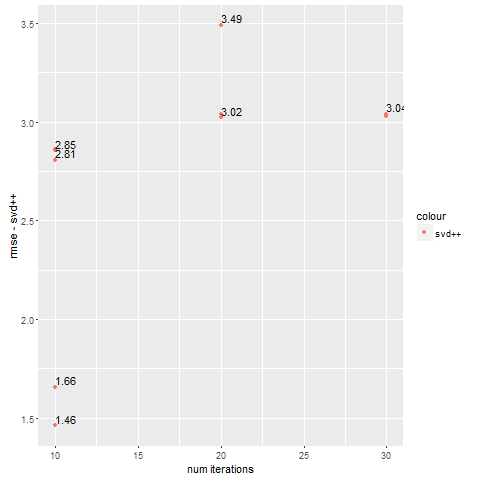
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| File name | userBased | svd | svd++ | Invalid results | hybrid |
| preferences\_1 | threshold: 0.05 | numFeatures: 2 lambda: 0.05 numIterations: 100 | numFeatures: 2 numIterations: 20 | If algorithm does not return result (returns NaN), get -2. If returns exception – get -1. |  |
| preferences\_2 | threshold: 0.05 | numFeatures: 2 lambda: 0.05 numIterations: 100 | numFeatures: 2 numIterations: 20 | If item based is used, get item avg, else get user avg. |  |
| preferences\_3 | threshold: 0.1 | numFeatures: 2 lambda: 0.1 numIterations: 100 | numFeatures: 2 numIterations: 10 | If item based is used, get item avg, else get user avg. |  |
| preferences\_3\_hybrid | threshold: 0.1 | numFeatures: 2 lambda: 0.1 numIterations: 100 | numFeatures: 2 numIterations: 10 | if item based is used, get item avg, else get user avg | 0.5\*userBased + 0.5 \* other |
| preferences\_3\_hybrid\_weighted | threshold: 0.1 | numFeatures: 2 lambda: 0.1 numIterations: 100 | numFeatures: 2 numIterations: 10 | if item based is used, get item avg, else get user avg | weighted: 0.6\*userBased + 0.4 \* other |
| preferences\_4 | threshold: 0.025 | numFeatures: 2 lambda: 0.025 numIterations: 100 | numFeatures: 2 numIterations: 30 | if item based is used, get item avg, else get user avg |  |
| preferences\_5 | threshold: 0.2 | numFeatures: 2 lambda: 0.05 numIterations: 50 | numFeatures: 3 numIterations: 20 | if item based is used, get item avg, else get user avg |  |
| preferences\_6 | threshold: 0.25 | numFeatures: 2 lambda: 0.05 numIterations: 150 | numFeatures: 3 numIterations: 30 | if item based is used, get item avg, else get user avg |  |
| preferences\_7 | threshold: 0.5 | numFeatures: 2 lambda: 0.1 numIterations: 100 | numFeatures: 2 numIterations: 10 | get avg of item avg and user avg |  |
| preferences\_8 | threshold: 0.1 | numFeatures: 5 lambda: 0.1 numIterations: 100 | numFeatures: 5 numIterations: 10 | get avg of item avg and user avg |  |

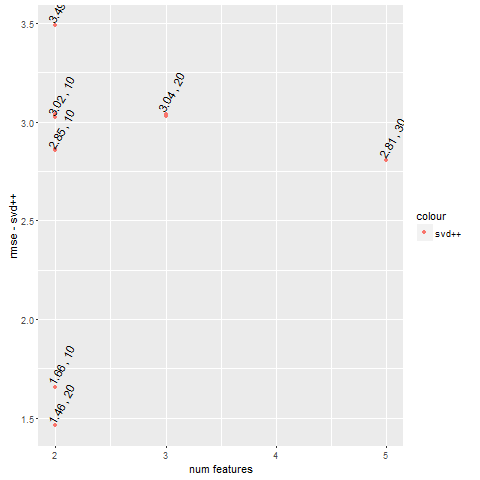
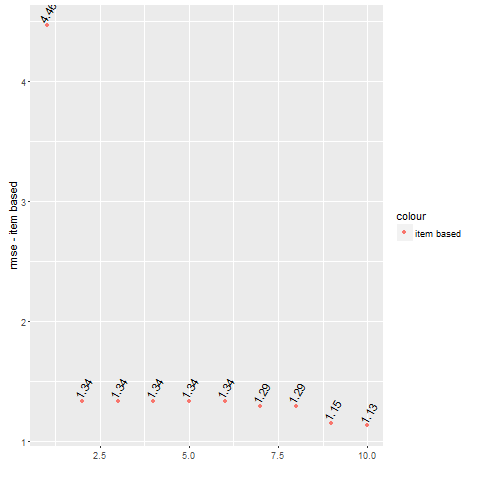
# Резултати

## Графики









## RMSE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| File name: | user based: | item based: | svd: | svd++: |
| preferences\_1 | train: 1.0410  test: 4.7702 | train: 1.4466  test: 4.4641 | train: 2.3610  train2: 0.7949  test: 2.2418 | train: 2.3002  train2: 3.4440  test: 3.4902 |
| preferences\_2 | train: 1.1281  test: 1.0949 | train: 1.5795  test: 1.3360 | train: 2.3296  train2: 0.7947  test: 1.4017 | train: 2.2153  train2: 3.4406  test: 3.0218 |
| preferences\_3 | train: 0.9950  test: 1.0948 | train: 1.4686  test: 1.3360 | train: 1.7947  train2: 0.8029  test: 1.2947 | train: 1.6787  train2: 3.2151  test: 2.8548 |
| preferences\_3\_hybrid | train: 1.0230  test: 1.0948 | train: 1.5111  test: 1.1489 | train: 1.9077  train2: 0.4014  test: 1.1135 | train: 1.7298  train2: 1.6135  test: 1.6571 |
| preferences\_3\_hybrid\_weighted | train: 1.0872  test: 1.0948 | train: 1.5787  test: 1.1281 | train: 1.8786  train2: 0.3215  test: 1.0980 | train: 1.4379  train2: 1.3015  test: 1.4604 |
| preferences\_4 | train: 1.1274  test: 1.0952 | train: 1.4988  test: 1.3360 | train: 3.4880  train2: 0.7928  test: 1.6725 | train: 2.5378  train2: 3.5034  test: 3.0412 |
| preferences\_5 | train: 1.1339  test: 1.0941 | train: 1.5411  test: 1.3360 | train: 2.0570  train2: 0.7957  test: 1.4030 | train: 2.1565  train2: 3.4188  test: 3.0403 |
| preferences\_6 | train: 1.1074  test: 1.0944 | train: 1.5931  test: 1.3360 | train: 2.5270  train2: 0.7930  test: 1.4187 | train: 2.4845  train2: 3.4682  test: 3.0311 |
| preferences\_7 | train: 1.1615  test: 1.0473 | train: 1.5152  test: 1.2899 | train: 1.8921  train2: 0.8041  test: 1.2889 | train: 1.6597  train2: 3.2112  test: 2.8612 |
| preferences\_8 | train: 1.0518  test: 1.0496 | train: 1.5910  test: 1.2899 | train: 1.9671  train2: 0.7514  test: 1.3011 | train: 1.6125  train2: 3.1994  test: 2.8058 |

# Бъдещо развитие

По – добро решение на *cold stars* проблема

* Например взимане на средното според категория

Тестване с други стойности за *threshold* с цел подобряване на *RMSE*.

# Използвана литература

* Идея: <https://www.kaggle.com/c/yelp-recsys-2013>
* Apache Mahout: <http://mahout.apache.org/>
* Описание на съществуващо решение: https://www.kaggle.com/c/yelp-recsys-2013/forums/t/5608/congratulate-to-the-25-winners/30932