RECSYS 2013 CHALLENGE

СИЯНА СЛАВОВА, 24963
ИВАН КАПУКАРАНОВ, 24958

Проблем

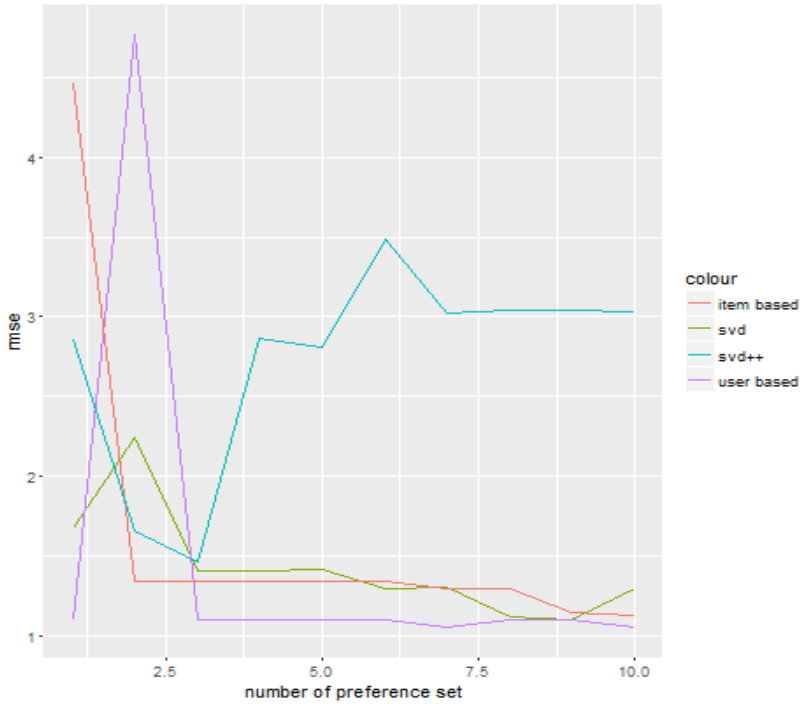
- Данни (предоставени от Yelp):
  - 10,000 фирми (business)
  - 8,000 check – in sites
  - 40,000 потребителя (users)
  - 200,000 ревюта (reviews)
- Задача:
  - Да се предскаже рейтинга(review), който потребител ***u*** би дал на продукт(business) ***i***.

Съществуващи решения

- На победителите в RecSys 2013 Challenge
  - Модели: *Matrix Factorization, Linear Regression, Regression Tree, Global Effects*.
  - Постигната RMSE: 1.21251

Нашето решение

- Apache Mahout (модели).
- R (RMSE, графики).
- Разделихме оригиналното обучаващо множество на две:
  - тестово(30 %)
  - обучаващо(70 %)



Предварителна обработка

- Конвертиране на ID-та от стринг в число
  - user_id
  - business_id
    - *Пример: "iUnAEpltJi0MCjmWrPu9w",43872*
- Опростяване на данните
  - Оставяне на user_id, business_id, review

Избрано решение

След проведените експерименти...

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	

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

...достигнахме до грешка (1.0473) при **user – based** подхода с **neighborhood threshold 0.5**.

Финално решение

- User based с threshold 0.5
- Ако алгоритъма не намери близки потребители спрямо този праг, за оценка се връща:
  - *(userAverage + businessAverage)/2*
- Cold start - средното за всички потребители или съответно фирми.

Submission резултати

93	↓19	gege	1.27585	15	Mon, 26 Aug 2013 09:35:57 (-0.1h)	
-		**СиянаСлавова**	**1.27585**	-	**Tue, 21 Jun 2016 19:56:50**	**Post-Deadline**

Post-Deadline Entry
If you would have submitted this entry during the competition, you would have been around here on the leaderboard.

94	↓8	Matteo Castagna	1.27598	14	Sat, 31 Aug 2013 21:19:36 (-22h)	

Бъдещо развитие

- По – добро решение на cold stars проблема
  - Например взимане на средното според категория
- Тестване с други стойности за threshold

Благодарим за вниманието!