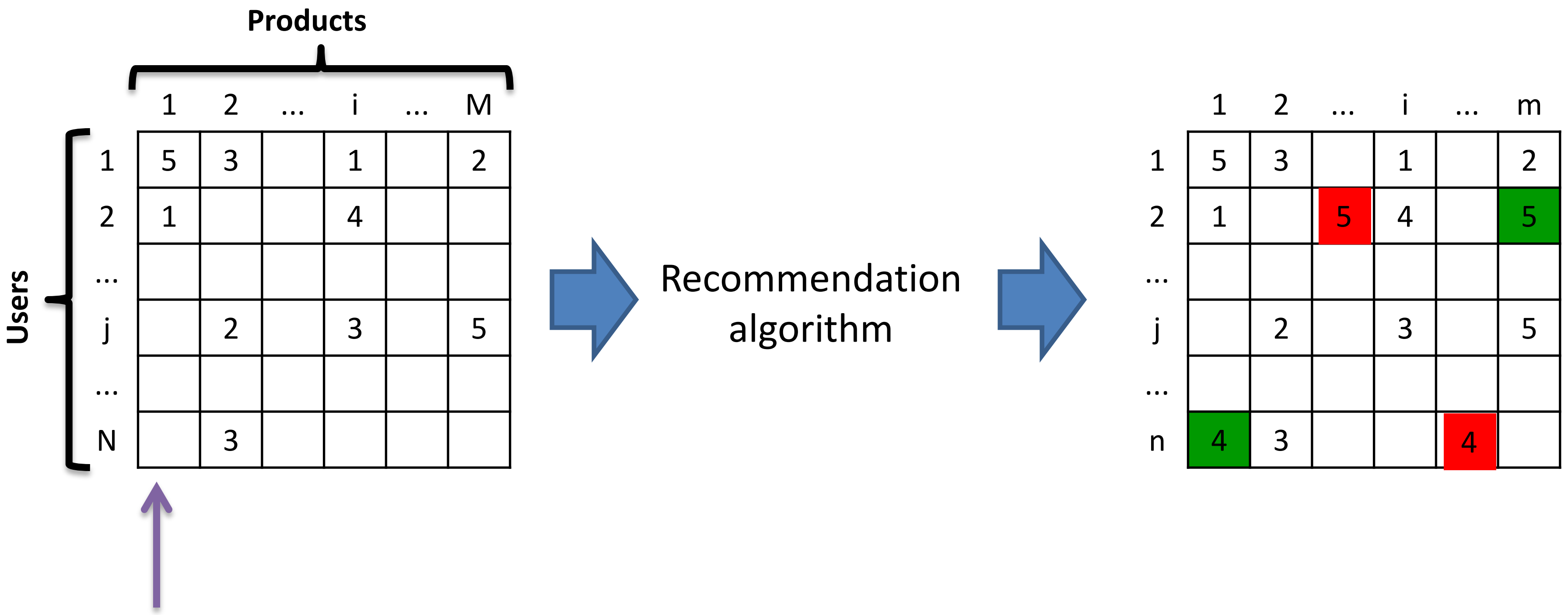
A Regularized Recommendation Algorithm with Probabilistic Sentiment-Ratings

Filipa Peleja, Pedro Dias and João Magalhães

Department of Computer Science
Faculdade de Ciências e Tecnologia
Universidade Nova de Lisboa

Objective: How to improve recommendations with user comments and reviews?



	1	2	…	i	…	M
1	5	3		1		2
2	1			4		
…						
j		2		3		5
…						
N		3				

	1	2	…	i	…	m
1	5	3		1		2
2	1		🟥 5	4		🟩 5
…						
j		2		3		5
…						
n	🟩 4	3			🟥 4	

Rating: ★★★★

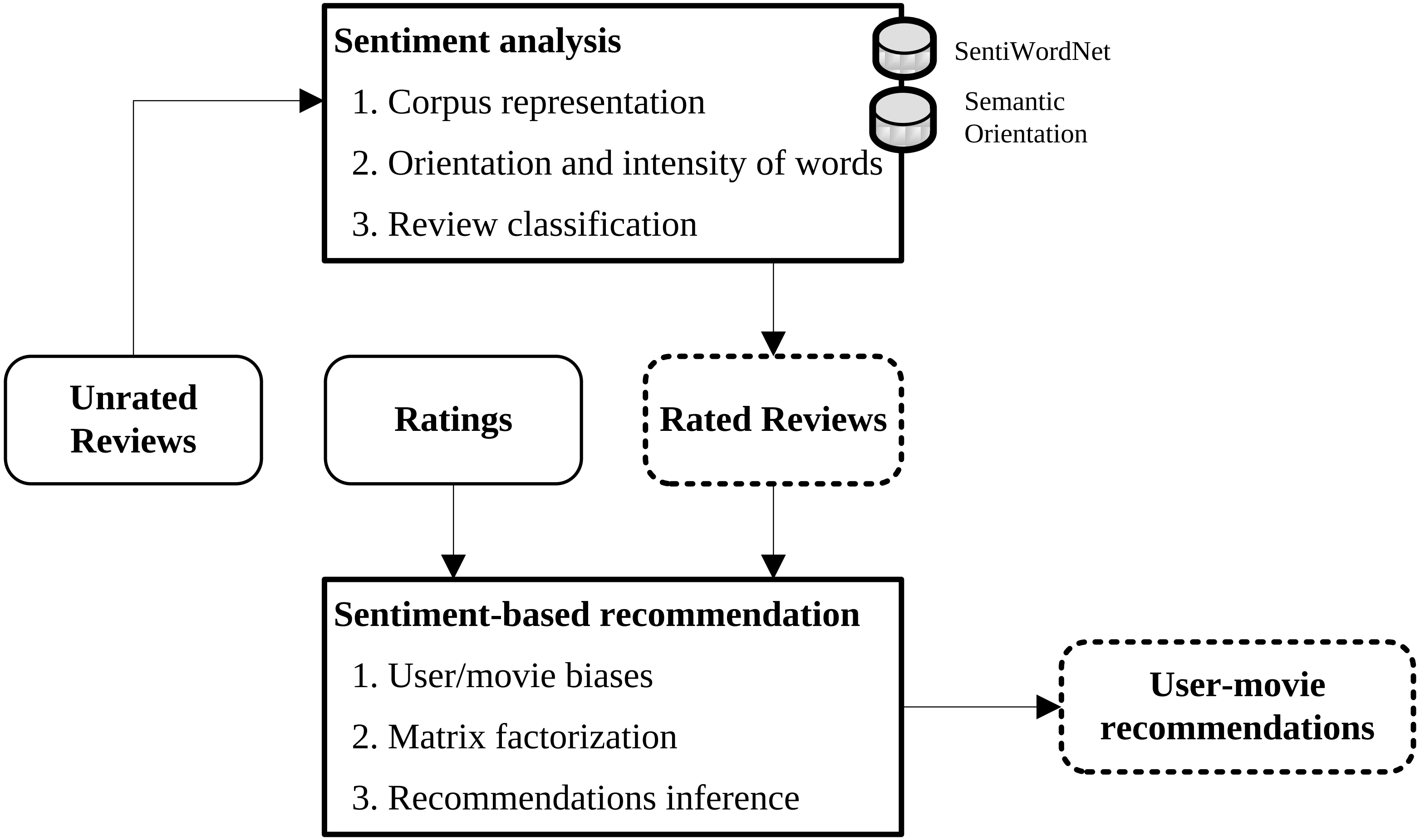
Review: Love it or hate it!

…

Rating: ★★

Review: This is a miserable film.

PROPOSED SOLUTION



Sentiment Analysis Challenges

- Opinions are written in natural language which implies :
  - subjectivity; - sarcasm; - irony; - idiomatic expressions; misspelling; etc.

- The same ***opinion word*** may be used in a positive or negative context

- Negative, Conditional and Comparative expressions

Opinion word orientation and intensity

- Semantic Orientation[1]:

$$SO(word) = \log_2\left(\frac{hits(word, "excellent")hits("poor")}{hits(word, "poor")hits("excellent")}\right)$$

- How positive or negative is an *opinion word*?
  - SentiWordNet[2]

[1] Turney, P. 2002, Thumbs up or thumbs down? Semantic Orientation applied to unsupervised classification of reviews

[2] Esuli, A. And Sebastiani, F., 2006, Sentiwordnet: A publicy available lexical resource for opinion mining

Example

“Love it or hate it.”

“However, can someone tell me what on earth the last page...”

word	family	SO (Google)	+ SentiWordNet	- SentiWordNet
love	n	-0.0824	1.375	0.0
it	nointerest	na	na	na
or	nointerest	na	na	na
hate	v	-0.8399	0.0	0.75
it	nointerest	na	na	na
however	r	-0.34153	0.5	0.5
someone	N	-0.65935	0.0	0.0
tell	V	-0.3956	0.875	0.625
me	nointerest	na	na	na
what	nointerest	na	na	na
on	nointerest	na	na	na
earth	n	-0.4041	0.0	0.625

Multiple Bernoulli Classification

$$p(ra_i = r \mid re_i) = \frac{f^r(ra_i, re_i)}{\sum_{L=1}^{10} f^L(ra_i, re_i)}$$

A classifier is learned for every rating value Thus, for each rating value there is a prediction for each review

The prediction is normalized accordingly to the predictions of all ratings

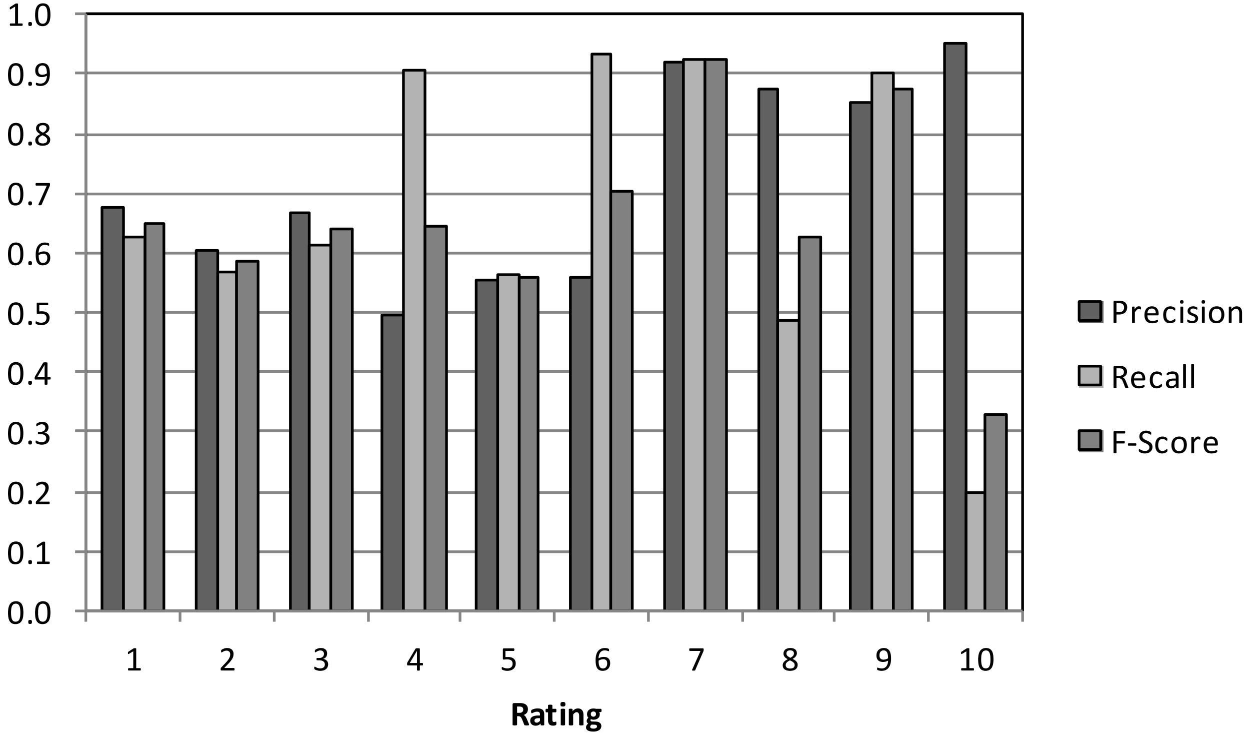
Rating range IMDb dataset is 1 to 10

Dataset
Reviews from IMDb

A total of 1,729,293 reviews were collected

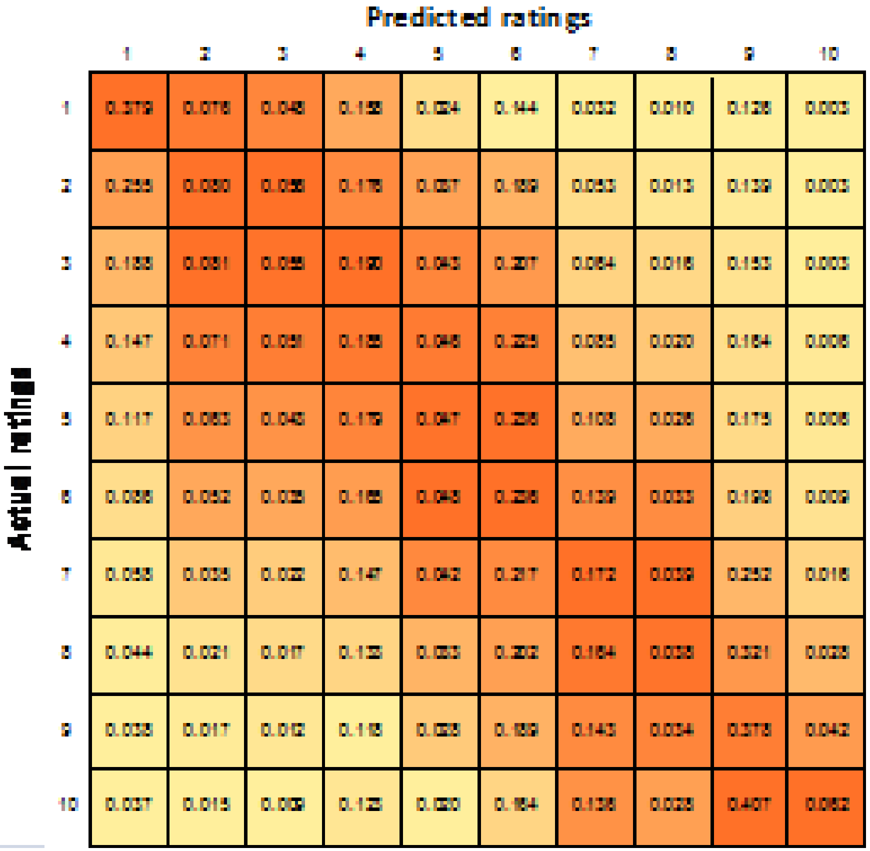
Split	#Reviews	Description
A	335,975	Only to train SA
B	335,975	Test SA/Train RS
C	417,147	Train RS (**no explicit ratings**)
D	335,976	Train RS
E	201,586	Test RS
F	102,634	Validate RS

Performance - Sentiment Analysis



F-score on the IMDb corpus

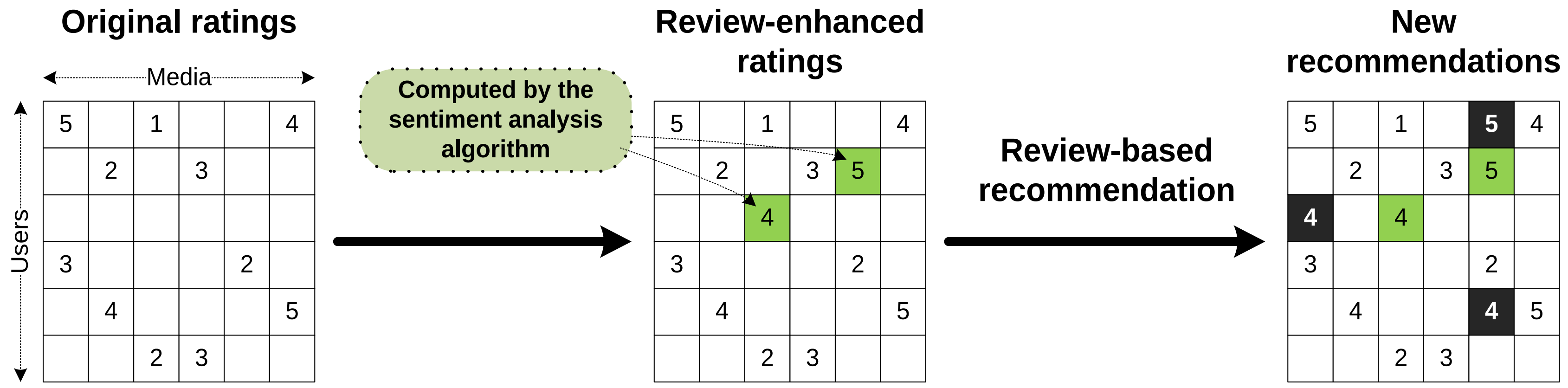
Performance - Sentiment Analysis



Predicted ratings

Actual ratings	1	2	3	4	5	6	7	8	9	10
1	0.379	0.076	0.048	0.158	0.024	0.144	0.032	0.010	0.128	0.003
2	0.255	0.090	0.058	0.176	0.037	0.169	0.053	0.013	0.139	0.003
3	0.188	0.081	0.059	0.190	0.043	0.207	0.064	0.016	0.153	0.003
4	0.147	0.071	0.051	0.185	0.046	0.225	0.085	0.020	0.164	0.006
5	0.117	0.063	0.043	0.179	0.047	0.236	0.108	0.026	0.175	0.006
6	0.086	0.052	0.035	0.165	0.046	0.236	0.139	0.033	0.198	0.009
7	0.058	0.035	0.022	0.147	0.042	0.217	0.172	0.039	0.252	0.016
8	0.044	0.021	0.017	0.125	0.035	0.202	0.164	0.038	0.321	0.028
9	0.038	0.017	0.012	0.118	0.028	0.169	0.143	0.034	0.378	0.042
10	0.037	0.015	0.009	0.125	0.020	0.164	0.136	0.028	0.407	0.062

Inferred Ratings in Recommendation Algorithm



Original ratings

5		1			4
	2		3		
3				2	
	4				5
		2	3		

Computed by the sentiment analysis algorithm

Review-enhanced ratings

5		1			4
	2		3	5	
		4			
3				2	
	4				5
		2	3		

Review-based recommendation

New recommendations

5		1		5	4
	2		3	5	
4		4			
3				2	
	4			4	5
		2	3		

Rating Matrix

$$R_{ra} = \begin{bmatrix} r_{11} & \cdots & r_{1m} \\ \vdots & \ddots & \vdots \\ r_{n1} & \cdots & r_{nm} \end{bmatrix} \longleftarrow$$

HIGHLY INCOMPLETE SINCE MOST ELEMENTS ARE EMPTY

PREDICT AN UNKNOWN RATING:

$$\hat{r}_{ui} = p_u . q_i$$

Users and Products represented in the same Latent Factor Space

With a SVD decomposition the Rating matrix

$$R_{ra} = \begin{bmatrix} u_{11} & \cdots & u_{1m} \\ \vdots & \ddots & \vdots \\ u_{n1} & \cdots & u_{nm} \end{bmatrix} . \begin{bmatrix} p_{11} & \cdots & p_{1m} \\ \vdots & \ddots & \vdots \\ p_{n1} & \cdots & p_{nm} \end{bmatrix}^T = P.Q^T$$

Matrix factorization enables the assessment of users preferences regarding the products by calculating their factor representations

Ratings matrix R_{ra}

Factorization with biases

Goal: Minimize the prediction Error

$$[P,Q] = \underset{p_u,q_i}{\arg\min} \sum_{r_{ui}\in R_{ra}} (r_{ui} - \hat{r}_{ui})^2 + \lambda(|| \, p_u \, ||^2 + || \, q_i \, ||^2 + b_u{}^2 + b_i{}^2)$$

Ratings matrix R_{ra}

Factorization with biases

$$[P,Q] = \underset{p_u,q_i}{\arg\min} \sum_{r_{ui}\in R_{ra}} (r_{ui} - \hat{r}_{ui})^2 + \sum_{r_{ui}\in \hat{R}_{rev}} (\hat{c}_{ui} - \hat{r}_{ui})^2 + \lambda(\| p_u \|^2 + \| q_i \|^2 + b_u{}^2 + b_i{}^2)$$

Ratings Inferred from the Sentiment Analysis Framework are Given to the RS
The Reviews Actual rating are Known

Ratings matrix *Rra* *factorization with sentiment-based* *regularization*

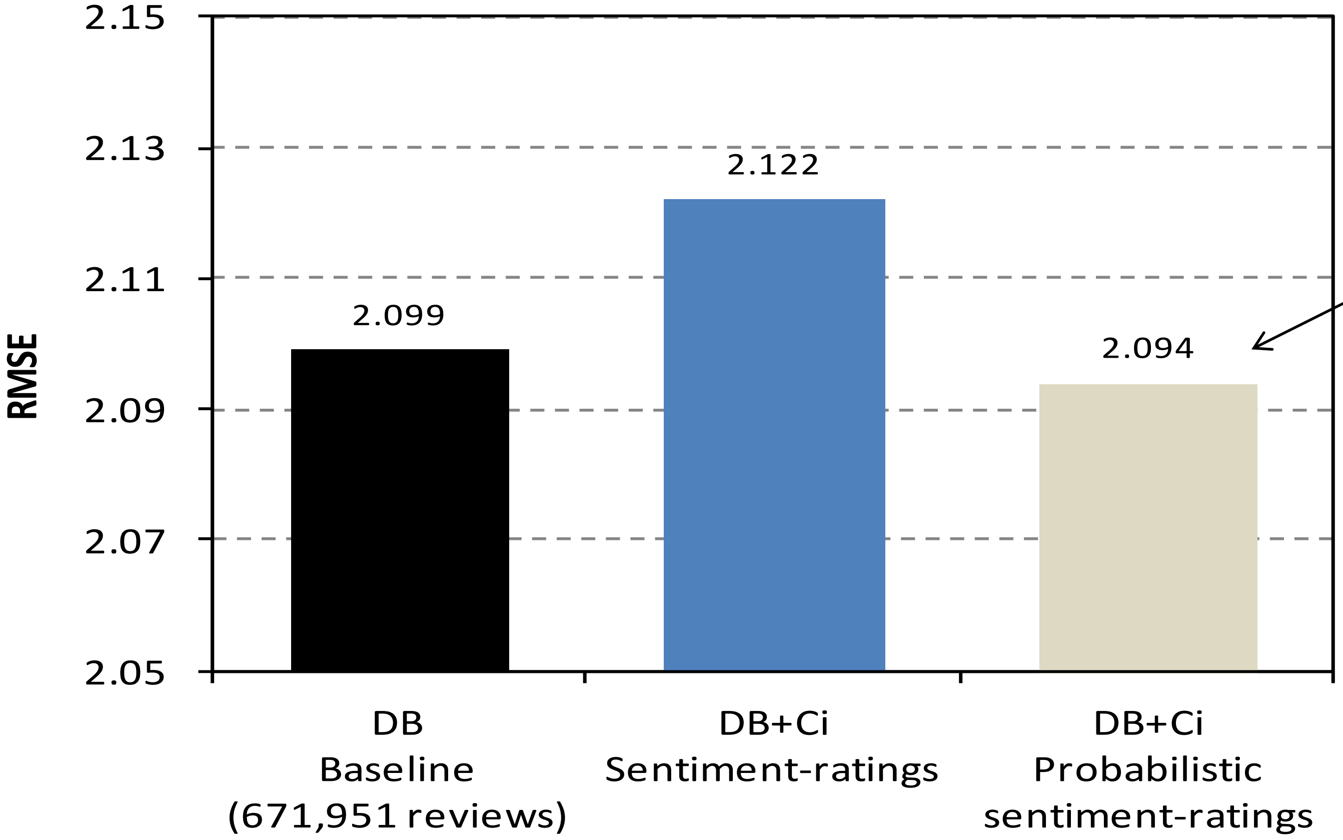
$$\mathrm{R} = \quad R_{ra}, \mathrm{R}_{rev}$$

Enrich the matrix R with ratings inferred from reviews with **known** and **Unknown Explicit Ratings** **Rrev**

$$\hat{R}_{ra} = \arg\min_{\hat{r}_{ui}} \sum_{r_{ui} \in R_{ra}} (r_{ui} - \hat{r}_{ui})^2 + \sum_{c_{ui} \in \hat{R}_{rev}} \theta_{ui} \cdot (\hat{c}_{ui} - \hat{r}_{ui})^2$$
$$+ \lambda(||\, p_u \,||^2 + ||\, q_i \,||^2 + b_u{}^2 + b_i{}^2),$$

The confidence level is given by de Sentiment Analysis framework

Recommendations: IMDb dataset



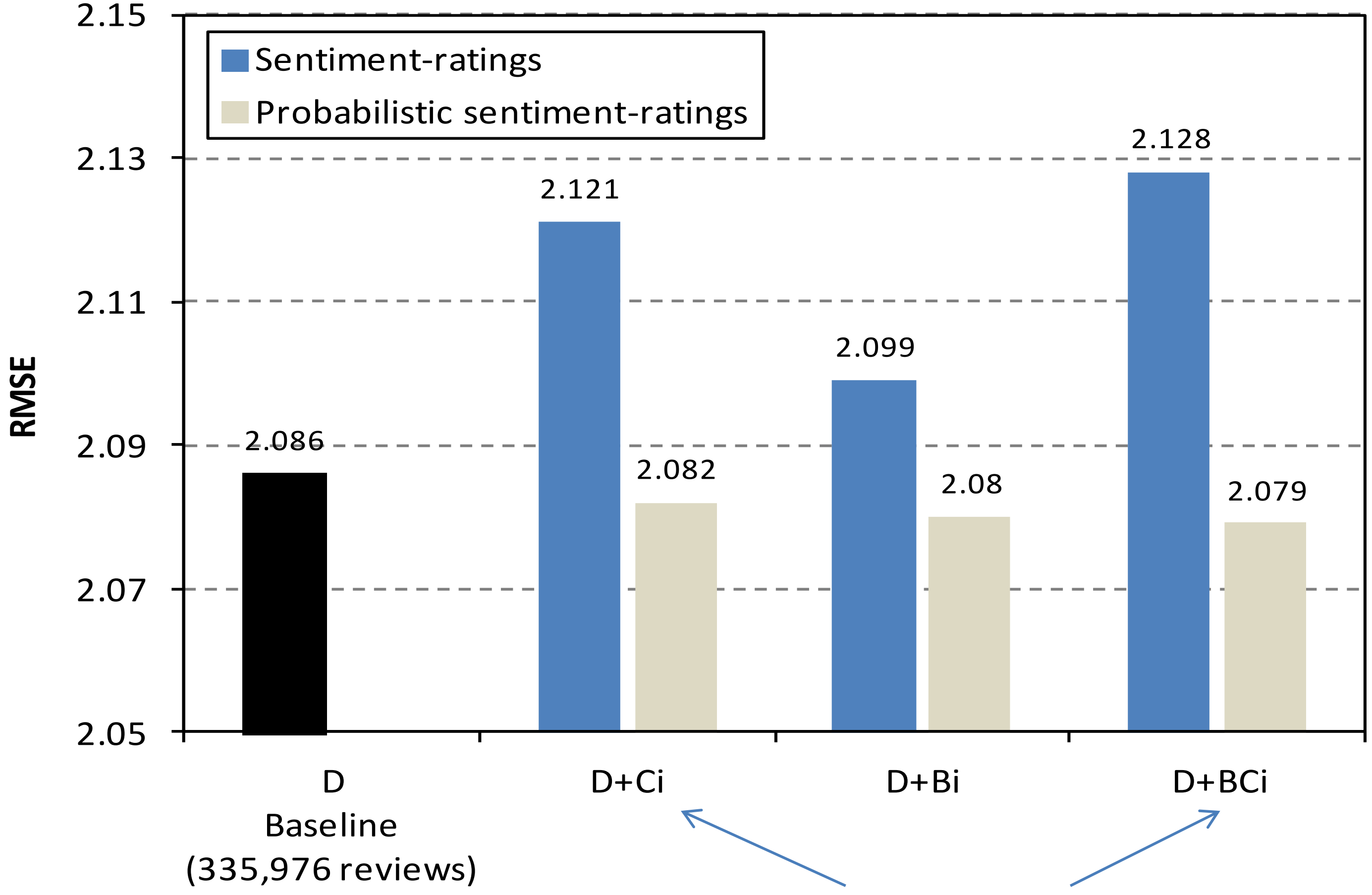
Lower RMSE error

Only Explicit Ratings

Inferred Ratings

Probabilistic Inferred Ratings

RECOMMENDATIONS: IMDB DATASET



Reviews with
Unknown ratings

Summary

- **Achievements**:
  - Extraction and sentiment analysis of users reviews
  - Introduced sentiment-based ratings in a recommendation algorithm

- **Next step**:
  - alternatives to SentiWordNet
  - semantic orientation metric
  - improve algorithm with opinion targets information

Thank you for your attention

Questions?