Factorization Meets the Neighborhood: a Multifaceted Collaborative Filtering Model

Yehuda Koren
AT & T Labs – Research

2008

Present by

Hong Ge
Sheng Qin

Info about paper & data-set

Factorization Meets the Neighborhood:
a Multifaceted Collaborative Filtering Model

- ACM Transactions on Knowledge Discovery from Data (TDD) archive
- Year of Publication: 2007; cited by 43 times
- Winner of the $1 Million Netflix Prize (2007)!!!!!
  - •9.34% improvement over the original Cinematch accuracy level
- Netflix data:
  - •Over 480,000 users, 17,770 movies
  - •Over 1 million observed ratings, 1% in total
  - •Rating: integer from 1 to 5 (with rating time-stamp)
  - •Multivariate, Time-Series

Title interpretation

Factorization Meets the Neighborhood:
a Multifaceted Collaborative Filtering Model

- Technique about **recommender systems**
- Based on: **Collaborative Filtering (CF)**
  - A process often applied to recommender systems
- Using: **Neighborhood Model** & **Latent Factor Model**
  - Two main disciplines of CF
- Solution: **Some amazing improvement** & **integration**
  - Innovative point of this paper

Background

Collaborative Filtering

Analyze past transactions to establish connections between users and movies.

- Relies on past user behavior
- Does not require explicit profile

Existing methods

Neighborhood

- Computing relationships between movies, or between users
- Not user → movie, but movie → movie

Latent factor

- Characterize user → movie on factors
- Factors are inferred from user feedback

- Why integrate?

The integrated model-why?

Neighborhood Models

- Estimate unknown ratings by using known ratings made by user for similar movies
- Good at capturing localized information
- Intuitive and simple to implement

Latent Factor Models

- Estimate unknown ratings by uncover latent features that explain known ratings
- Efficient at capturing global information

The integrated model-why?

Reasons:

- Neighborhood Model: Good at capture localized information
- Latent Factor Model: Efficient at capturing global information
- Neither is able to capture all information
- Complementary with each other.
- Not account implicit feedback
- It's not tried before, why not?

The integrated model-how?

- **How?**
  - Sum the predications of revised Neighborhood Model(NewNgbr) and revised Latent Model (SVD++)
- **Some details**
  - I guess you may want take a nap now.
  - Just joking!

Some background before we go further

The Netflix data

- Many items in this matrix are missing
- Need find a good estimate for (most of efforts are dealing with this!)

$$
\begin{array}{cc}
 & \text{Ratings} \\
\text{Users} & \left\{\begin{array}{cccc}
r_{11} & r_{12} & \cdots & r_{1i} \\
r_{21} & r_{22} & \cdots & r_{2i} \\
\cdots & \cdots & \cdots & \cdots \\
r_{u1} & r_{u2} & \cdots & r_{ui}
\end{array}\right.
\end{array}
$$

[Netflix data]

Baseline estimates

- μ is the average rating over all movies
- b_u, b_j indicate the observed deviations of user *u* and item *I, respectively, from the average*

$$b_{ui} = \mu + b_u + b_i$$

[baseline estimator]

Neighborhood Model

- Estimate $\hat{r}_{ui}$ by using known ratings made by user for similar movies:

User specific weights

$$\hat{r}_{ui} = b_{ui} + \sum_{j \in S^k(i;u)} w_{uj} \times (r_{uj} - b_{uj})$$

k most similar movies rated by u, also known as Neighbors

Neighborhood models- Revised

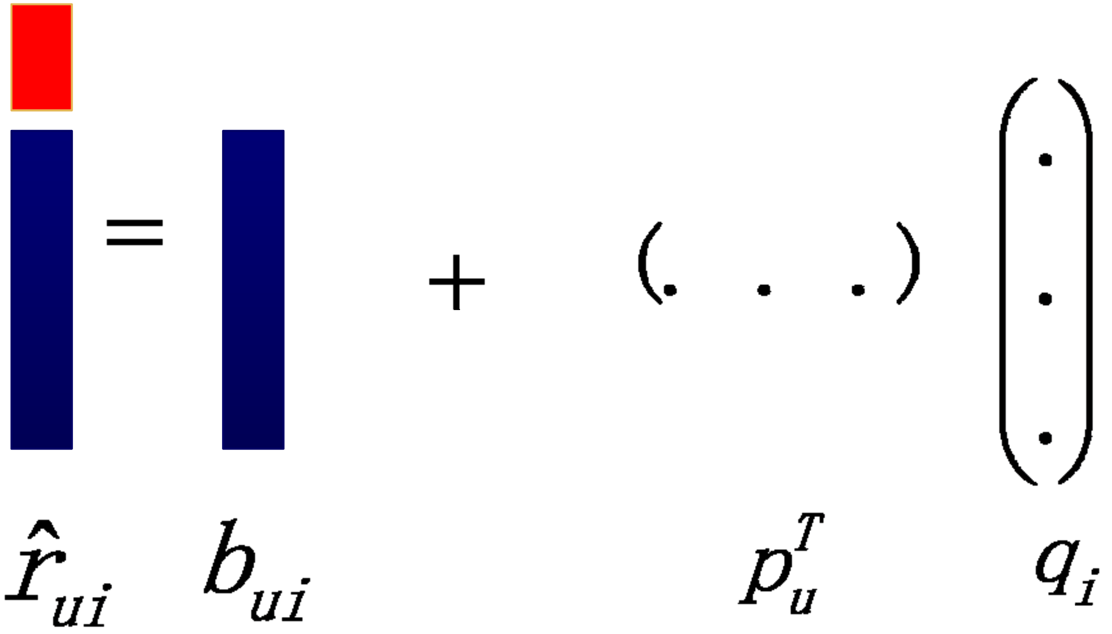
❖ New Neighborhood model:

- introduce implicit feedback effect
- use global rather than user-specific weights

❖ New predicting rule:

$$\hat{r}_{ui} = b_{ui} + \boxed{R^{-1/2} \sum_{j \in R^k(i;u)} w_{ij}} \, (r_{uj} - b_{uj}) + \boxed{N^{-1/2} \sum_{j \in N^k(i;u)} c_{ij}}$$

- **Estimate $\hat{r}_{ui}$ by uncover latent features that explain observed ratings:**



  - p_u, q_i are user-factors vector and item-factors vector respectively

Latent Model- Revised

❖ Introduce implicit feedback information

- Asymmetric-SVD

$$\hat{r}_{ui} = \underbrace{b_{ui}}_{\text{baseline estimate}} + q_i^T \Big(R^{-1/2} \sum_{j \in R(u)} (r_{uj} - b_{uj}) + \underbrace{N^{-1/2} \sum_{j \in N(u)} y_j}_{\text{Implicit feedback effect}} \Big)$$

❖ SVD++

- No theoretical explanation, it just works!

$$\hat{r}_{ui} = b_{ui} + q_i^T \Big(p_u + N^{-1/2} \sum_{j \in N(u)} y_j \Big)$$

- This model will be integrated with Neighborhood Model

How well does it work?

- Here is the result.

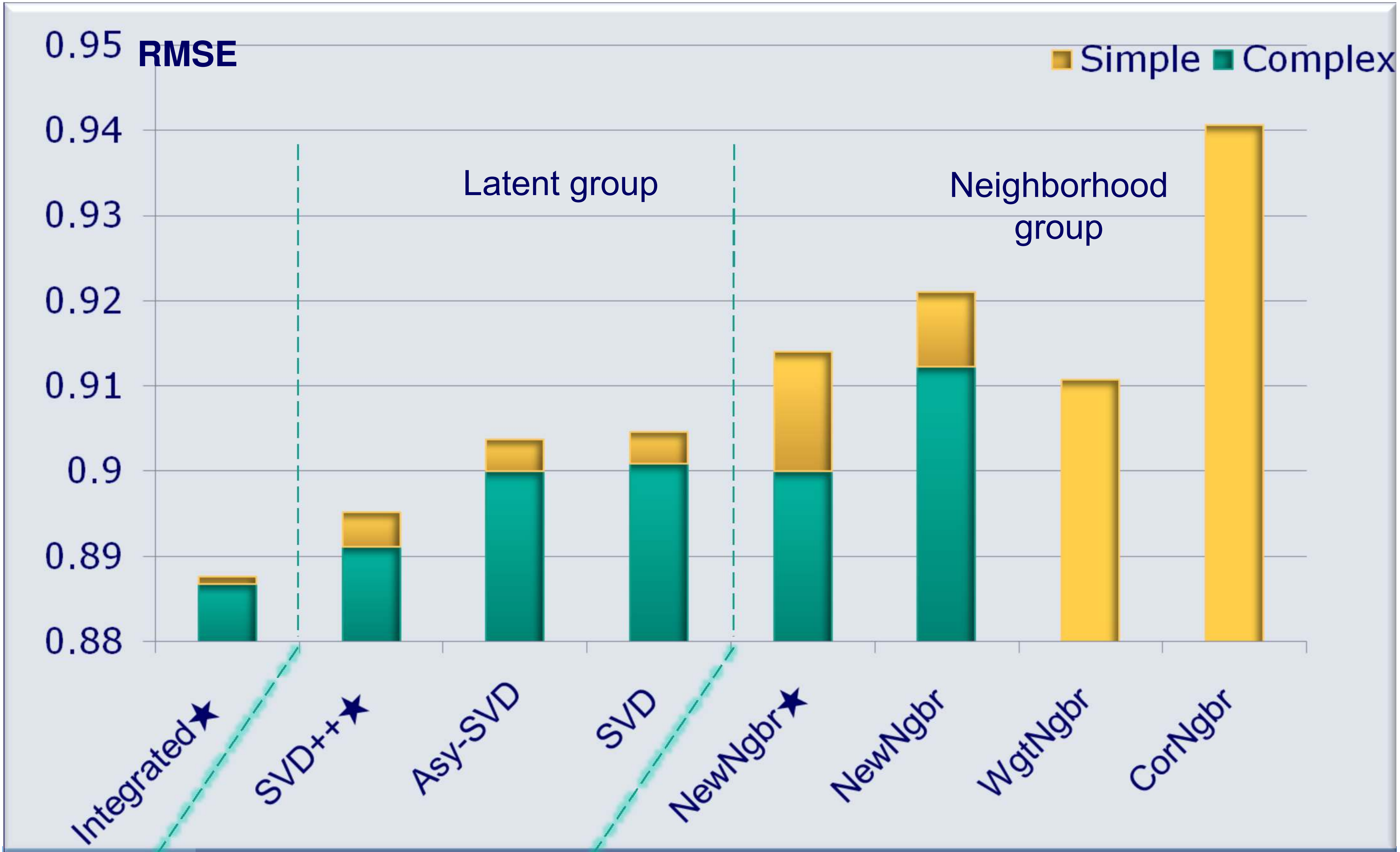
Test (Instructions)

Measured by Root Mean Square Error (RMSE)

$$\sqrt{\sum_{(u,i)\in TestSet}(r_{ui}-\hat{r}_{ui})^2/|TestSet|}$$

Abbreviation instructions	
Integrated★	Proposed Integrated Model
SVD++★	Proposed improved Latent Factor
SVD	Common Latent Factor
New Ngbr★	Proposed neighborhood, with implicit feedback
New Ngbr	Proposed neighborhood, without implicit feedback
WgtNgbr	improved neighborhood of the same user
CorNgbr	Popular neighborhood method

Experimental results —— RMSE



Time cost

NewNeighborhood

Time*(min)	10	27	58
Neighbors	250	500	Infinity
Precision	0.9014	-0.0010	-0.0004

SVD++

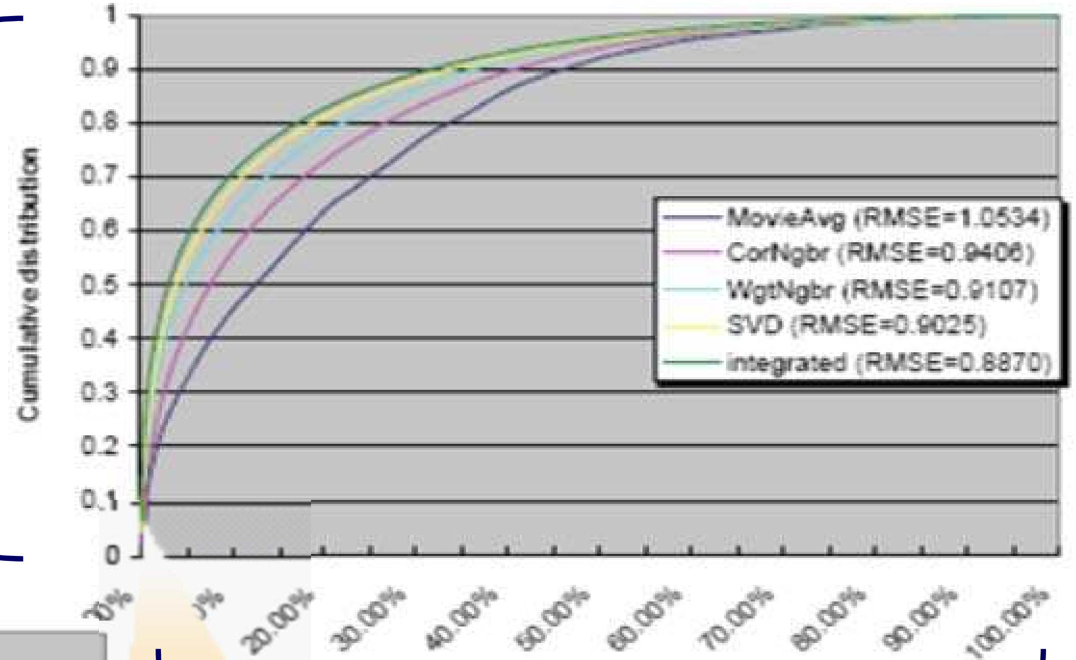
Time*(min)	--	--	--
Factors	50	100	200
Precision	0.8952	-0.0028	-0.0013

Integrated

Time(min)	17	20	25
Neighbors	300	300	300
Factors	50	100	200
Precision	0.8877	-0.0007	-0.0002

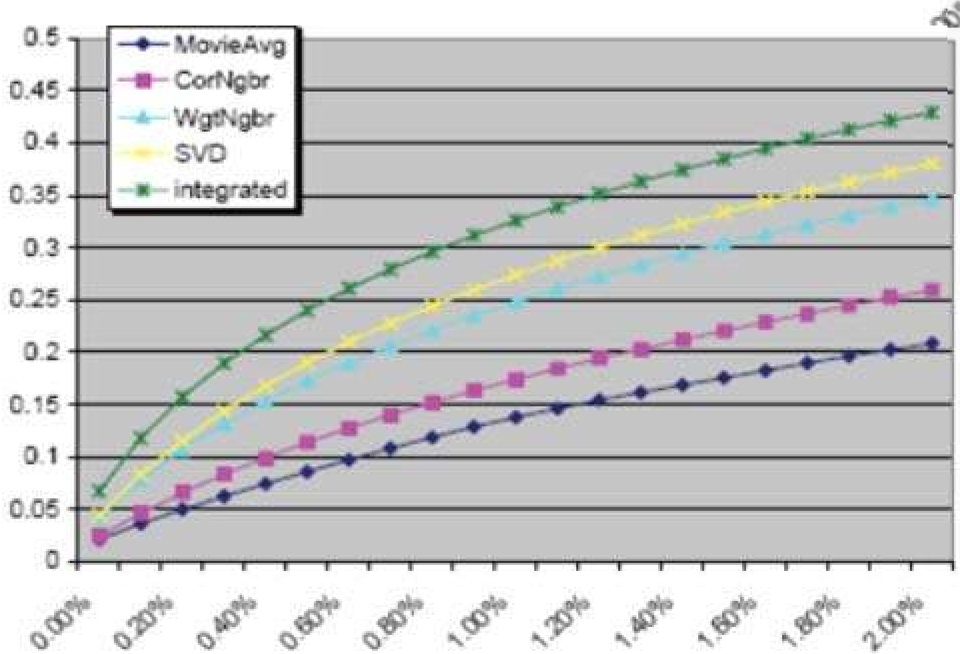
Experimental results —— top K

Y axis:
Probability distribution of the observed best movie returned

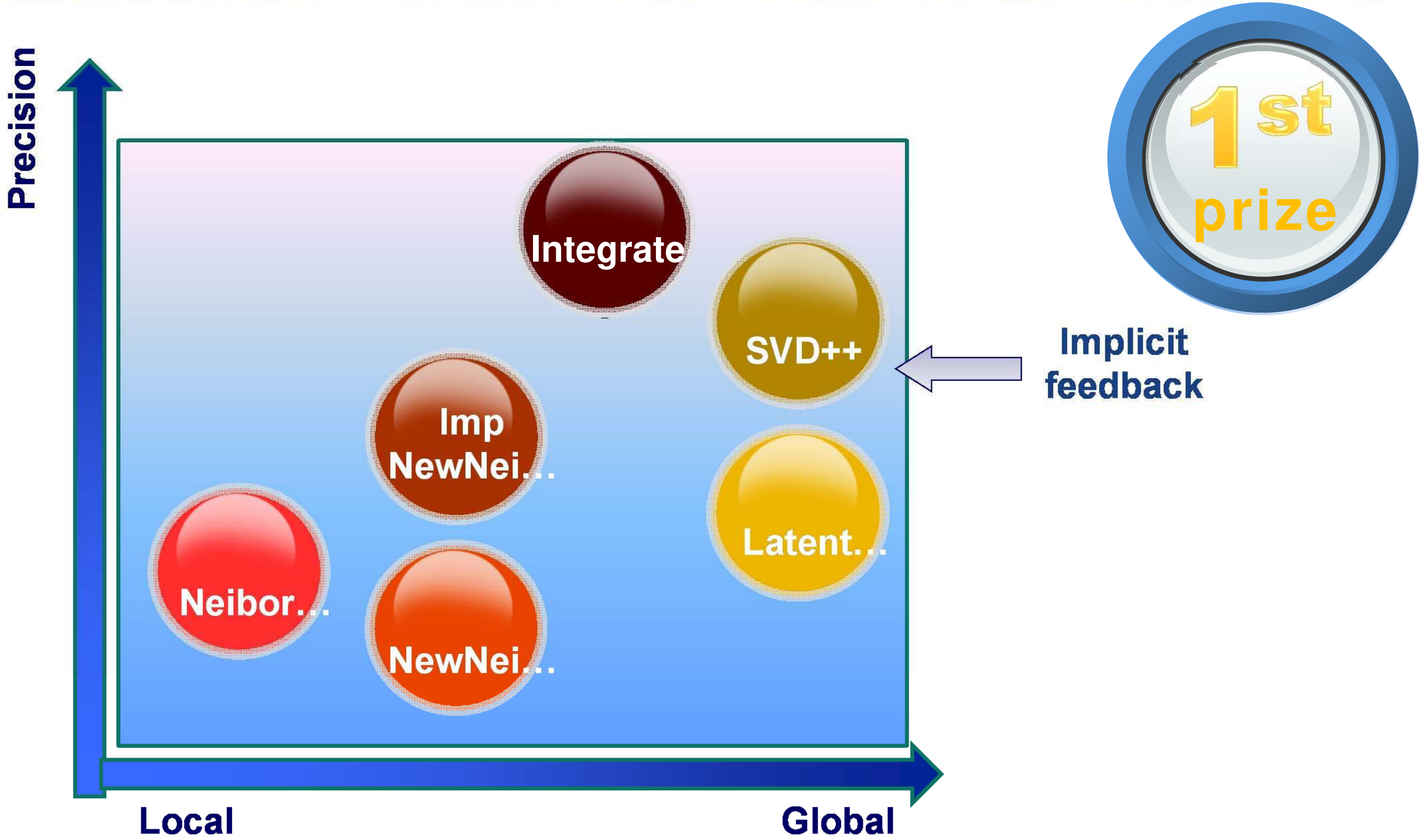


0%~2%

X axis:
Threshold of return in percentile



Conclusion



Precision
Integrate
SVD++
Imp NewNei...
Latent...
Neibor...
NewNei...
Local
Global
Implicit feedback
1st prize

Hard to beat, but…

- Ignored time-stamps
  - •Time-stamps available (from 1998 to 2005)
  - •Temporal dynamics matters

Example 1

6 years later…

Action

Romance

Hard to beat, but...

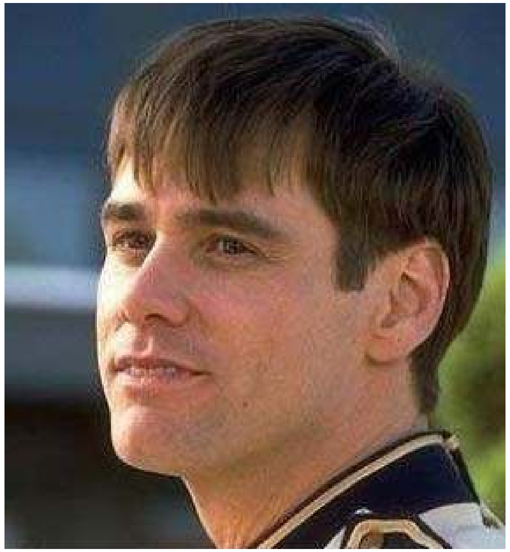
- Ignored time-stamps
  - •Time-stamps available (from 1998 to 2005)
  - •Temporal dynamics matters

Example 2

5 5 5 5 5 5 5

2 days later…

3 2 4 3 3 4



Hard to beat, but…

- Temporal dynamics are too personal

•Represented in author’s latest publication, with comparison
•May move the model towards local level

References

- Yehuda Koren, Factorization meets the neighborhood: a multifaceted collaborative filtering model, in Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining (Las Vegas, Nevada, USA: ACM, 2008), 426-434

- Yehuda Koren, The BellKor Solution to the Netflix Grand Prize, August 2009

- **Questions?**