Predicting Performance in Recommender Systems

Alejandro Bellogín
Supervised by Pablo Castells and Iván Cantador
Escuela Politécnica Superior
Universidad Autónoma de Madrid

@abellogin
alejandro.bellogin@uam.es

Motivation

Is it possible to predict the accuracy of a recommendation?

Hypothesis

Data that are commonly available to a Recommender System could contain **signals** that enable an *a priori* estimation of the **success** of the recommendation

Research Questions

1. Is it possible to define a performance prediction **theory** for recommender systems in a sound, formal way?
2. Is it possible to **adapt** query performance techniques (from IR) to the recommendation task?
3. What kind of **evaluation** should be performed? Is IR evaluation still valid in our problem?
4. What kind of recommendation **problems** can these models be applied to?

Predicting Performance in Recommender Systems

RQ1. Is it possible to define a performance prediction **theory** for recommender systems in a sound, formal way?

a) Define a predictor of performance $\gamma = \gamma(u, i, r, \ldots)$

b) Agree on a performance metric $\mu = \mu(u, i, r, \ldots)$

c) Check predictive power by measuring correlation

$$\mathrm{corr}([\gamma(x_1), \ldots, \gamma(x_n)], [\mu(x_1), \ldots, \mu(x_n)])$$

d) Evaluate final performance: dynamic vs static

Predicting Performance in Recommender Systems

RQ2. Is it possible to **adapt** query performance techniques (from IR) to the recommendation task?

- In IR: "Estimation of the system's performance in response to a specific query"
- Several predictors proposed
- We focus on query clarity → **user clarity**

User clarity

- It captures uncertainty in user's data
  - Distance between the user's and the system's probability model

$$\text{clarity}(u) = \sum_{x \in X} p(x \mid u) \log \left(\frac{p(x \mid u)}{p_c(x)} \right)$$

user's model

system's model

  - *X* may be: users, items, ratings, or a combination

User clarity

- Three user clarity formulations:

Name	Vocabulary	User model	Background model
Rating-based	Ratings	$p(r \mid u)$	$p_c(r)$
Item-based	Items	$p(i \mid u)$	$p_c(i)$
Item-and-rating-based	Items rated by the user	$p(r \mid i, u)$	$p_{ml}(r \mid i)$

$$\text{clarity}(u) = \sum_{x \in X} p(x \mid u) \log \left(\frac{p(x \mid u)}{p_c(x)} \right)$$

user model

background model

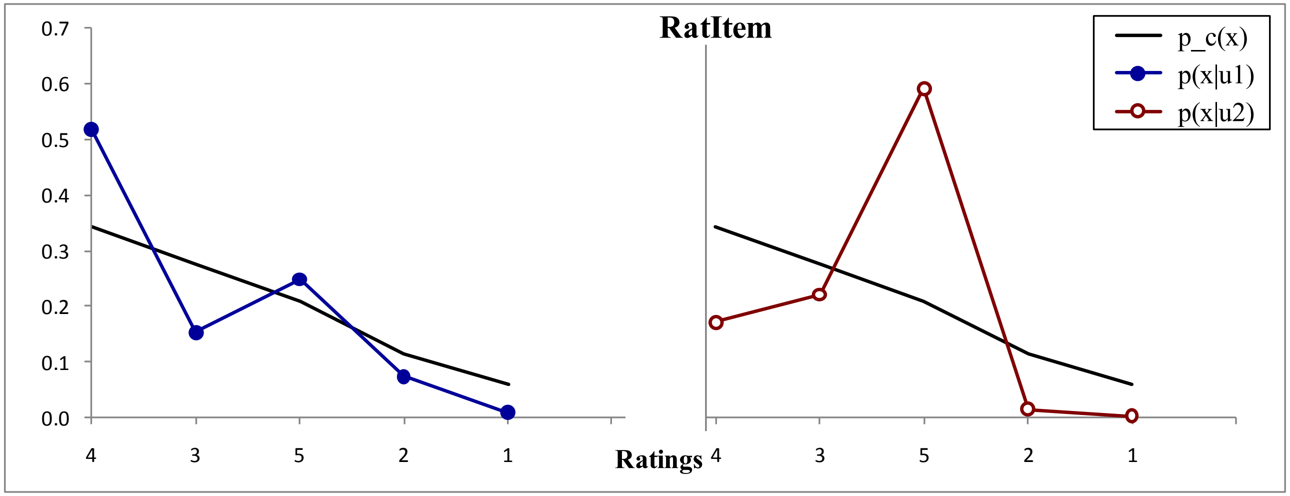
User clarity

- Seven user clarity models implemented:

Name	Formulation	User model	Background model
RatUser	Rating-based	$p_U(r \mid i,u);\ p_{UR}(i \mid u)$	$p_c(r)$
RatItem	Rating-based	$p_I(r \mid i,u);\ p_{UR}(i \mid u)$	$p_c(r)$
ItemSimple	Item-based	$p_R(i \mid u)$	$p_c(i)$
ItemUser	Item-based	$p_{UR}(i \mid u)$	$p_c(i)$
IRUser	Item-and-rating-based	$p_U(r \mid i,u)$	$p_{ml}(r \mid i)$
IRItem	Item-and-rating-based	$p_I(r \mid i,u)$	$p_{ml}(r \mid i)$
IRUserItem	Item-and-rating-based	$p_{UI}(r \mid i,u)$	$p_{ml}(r \mid i)$

User clarity

- Predictor that captures uncertainty in user's data
- Different formulations capture different nuances
- More dimensions in RS than in IR: user, items, ratings, features, …



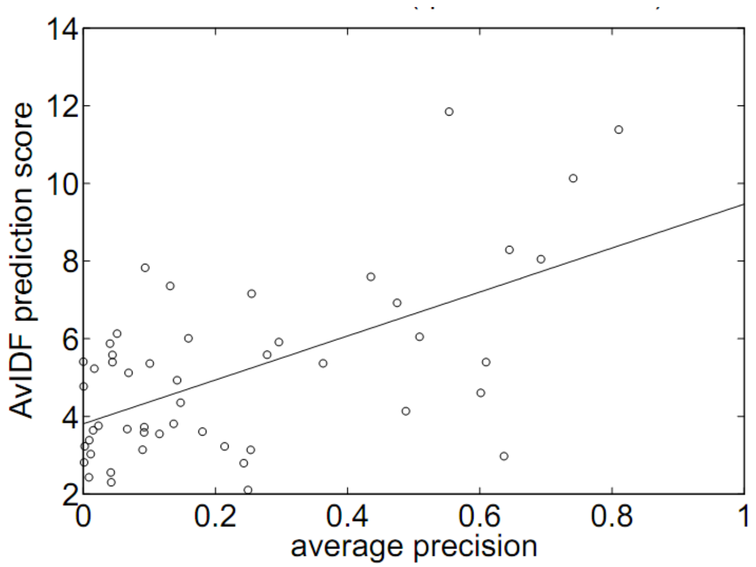
UNIVERSIDAD AUTONOMA DE MADRID

IRG IR Group @ UAM

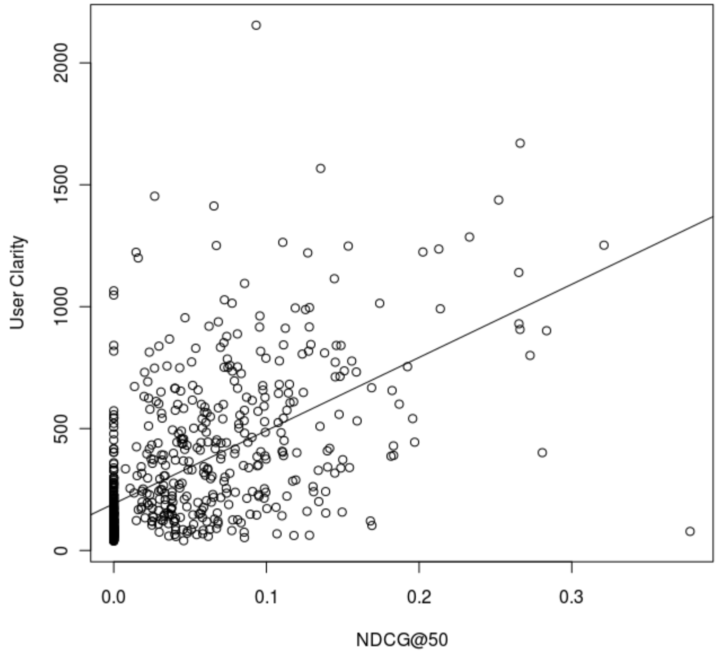
Predicting Performance in Recommender Systems

RQ3. What kind of **evaluation** should be performed? Is IR evaluation still valid in our problem?

- In IR: Mean Average Precision + correlation
  - 50 points (queries) vs 1000+ points (users)



r ~ 0.57



RQ3. What kind of **evaluation** should be performed? Is IR evaluation still valid in our problem?

- In IR: Mean Average Precision + correlation
  - 50 points (queries) vs 1000+ points (users)
- Performance metric is not clear: error-based, precision-based?
  - What is performance?
  - It may depend on the final application
- Possible bias
  - E.g., towards users or items with larger profiles

RQ4. What kind of recommendation **problems** can these models be applied to?

- Whenever a combination of strategies is available
- Example 1: dynamic neighbor weighting
- Example 2: dynamic ensemble recommendation

Dynamic neighbor weighting

- The user's neighbors are weighted according to their similarity
- Can we take into account the uncertainty in neighbor's data?

- User neighbor weighting [1]
  - Static: $$g(u,i) = C \sum_{v \in N[u]} \mathrm{sim}(u,v) \times rat(v,i)$$
  - Dynamic: $$g(u,i) = C \sum_{v \in N[u]} \gamma(v) \times \mathrm{sim}(u,v) \times rat(v,i)$$

Dynamic hybrid recommendation

- Weight is the same for every item and user (learnt from training)
- What about boosting those users predicted to perform better for some recommender?

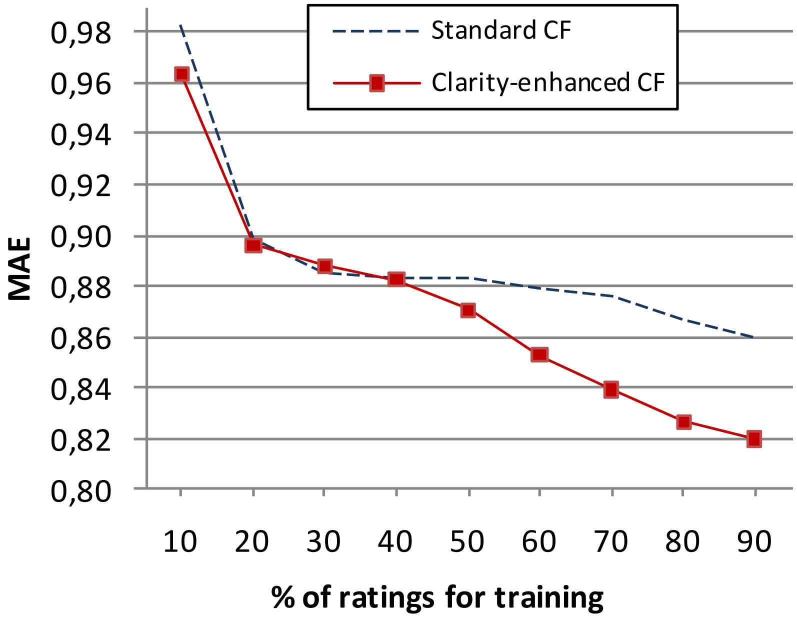
- Hybrid recommendation [3]
  - Static: $$g(u,i) = \lambda \times g_{R1}(u,i) + (1-\lambda) \times g_{R2}(u,i)$$
  - Dynamic: $$g(u,i) = \gamma(u) \times g_{R1}(u,i) + (1-\gamma(u)) \times g_{R2}(u,i)$$

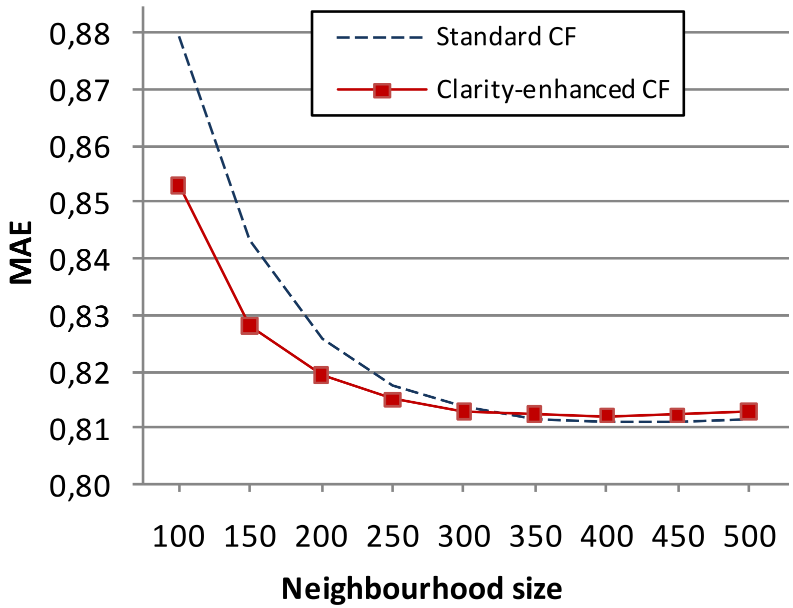
Results – Neighbor weighting

- Correlation analysis [1]
  - With respect to Neighbor Goodness metric: “how good a neighbor is to her vicinity”

% training	10%	20%	30%	40%	50%	60%	70%	80%	90%
correlation	-0.10	0.10	0.18	0.18	0.18	0.17	0.17	0.15	0.15

- Performance [1] (MAE = Mean Average Error, the lower the better)





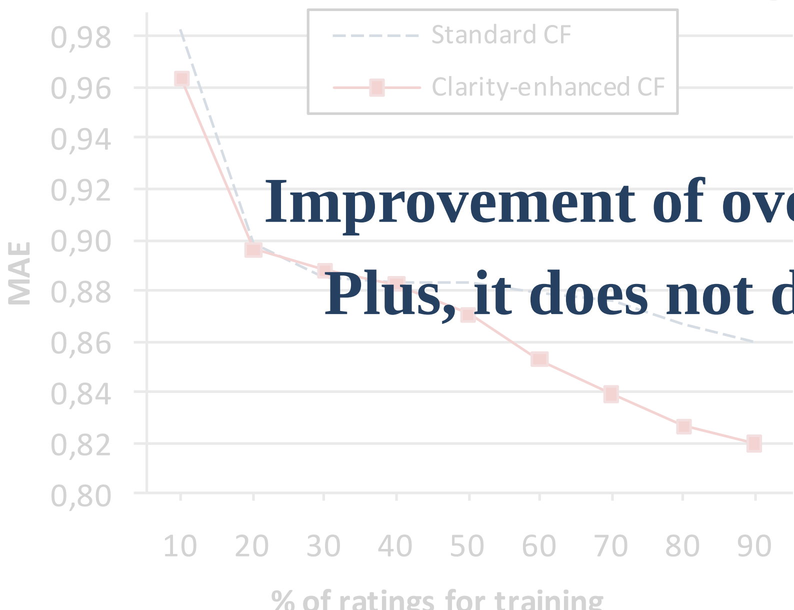
Results – Neighbour weighting

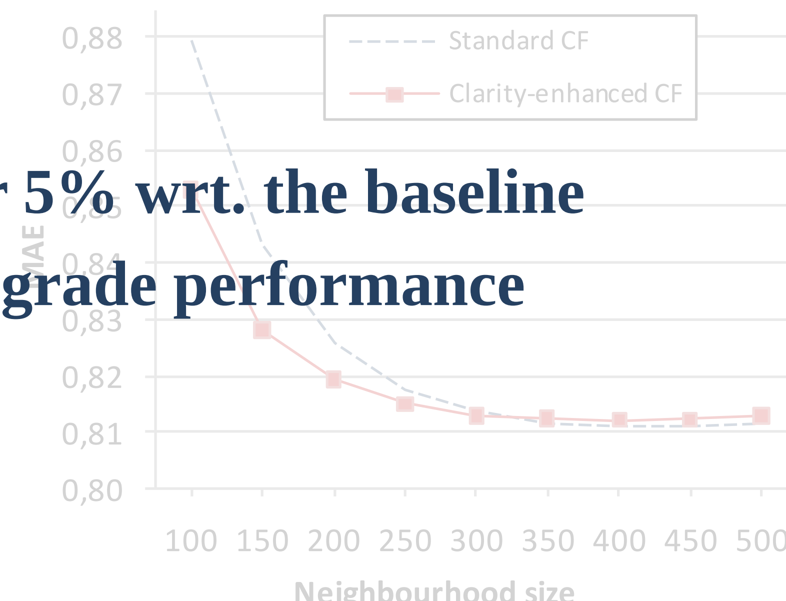
- Correlation analysis [1]
  - With respect to Neighbour Goodness metric: "how good a neighbour is to her vicinity"

Positive, although not very strong correlations

% training	10%	20%	30%	40%	50%	60%	70%	80%	90%
correlation	-0.10	0.10	0.18	0.18	0.18	0.17	0.17	0.15	0.15

- Performance [1] (MAE = Mean Average Error, the lower the better)





Improvement of over 5% wrt. the baseline

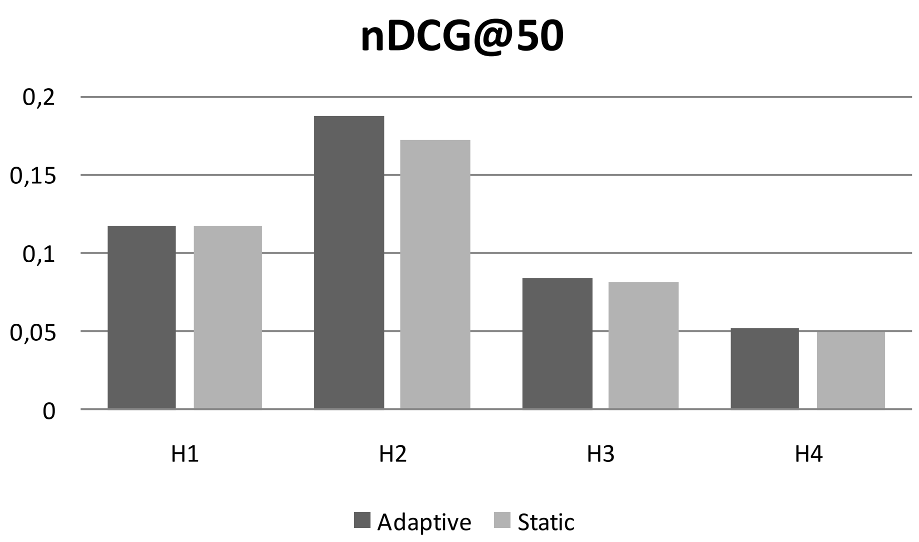
Plus, it does not degrade performance

Results – Hybrid recommendation

- Correlation analysis [2]
  - With respect to nDCG@50 (nDCG, normalized Discount Cumulative Gain)

Predictor	CBF	IB	TF-L1	TF-L2	UB	**Median**	**Mean**
ItemSimple	0.257	0.146	0.521	0.564	0.491	0.491	0.396
ItemUser	0.252	0.188	0.534	0.531	0.483	0.483	0.398
RatUser	0.234	0.182	0.507	0.516	0.469	0.469	0.382
RatItem	0.191	0.184	0.442	0.426	0.395	0.395	0.328
IRUser	0.171	-0.092	0.253	0.399	0.257	0.253	0.198
IRItem	0.218	0.152	0.453	0.416	0.372	0.372	0.322
IRUserItem	0.265	0.105	0.523	0.545	0.444	0.444	0.376

- Performance [3]



Results – Hybrid recommendation

- Correlation analysis [2]
  - With respect to nDCG@50 (nDCG, normalized Discount Cumulative Gain)

Predictor	CBF	IB	TF-L1	TF-L2	UB	Median	Mean
ItemSimple	0.257	0.146	0.521	0.564	0.491	0.491	0.396
ItemUser	0.252	0.188	0.534	0.551	0.483	0.483	0.398
RatUser	0.234	0.182	0.507	0.516	0.469	0.469	0.382
RatItem	0.191	0.184	0.442	0.426	0.395	0.395	0.328
IRUser	0.171	-0.092	0.253	0.399	0.257	0.253	0.198
IRItem	0.218	0.152	0.453	0.416	0.372	0.372	0.322
IRUserItem	0.265	0.105	0.523	0.545	0.444	0.444	0.376

In average, most of the predictors obtain positive, strong correlations

- Performance [3]

Dynamic strategy outperforms static for different combination of recommenders

Summary

- Inferring user's performance in a recommender system
- Different adaptations of query clarity techniques

- Building dynamic recommendation strategies
  - Dynamic neighbor weighting: according to expected goodness of neighbor
  - Dynamic hybrid recommendation: based on predicted performance

- Encouraging results
  - Positive predictive power (good correlations between predictors and metrics)
  - Dynamic strategies obtain better (or equal) results than static

Related publications

- [1] A Performance Prediction Aproach to Enhance Collaborative Filtering Performance. A. Bellogín and P. Castells. In ECIR 2010.
- [2] Predicting the Performance of Recommender Systems: An Information Theoretic Approach. A. Bellogín, P. Castells, and I. Cantador. In ICTIR 2011.
- [3] Performance Prediction for Dynamic Ensemble Recommender Systems. A. Bellogín, P. Castells, and I. Cantador. In press.

Future Work

- Explore other input sources
  - Item predictors
  - Social links
  - Implicit data (with time)

- We need a theoretical background
  - Why do some predictors work better?

- Larger datasets

FW – Other input sources

- Item predictors
- Social links
- Implicit data (with time)

- Item predictors could be very useful:
  - Different recommender behavior depending on item attributes
  - They would allow to capture popularity, diversity, etc.

FW – Other input sources

- Item predictors
- Social links
- Implicit data (with time)

- First results using social-based predictors
  - Combination of social and CF
  - Graph-based measures as predictors
  - “Indicators” of the user strength

	P@5			nDCG@5		
	H1	H2	H3	H1	H2	H3
Average Neigh Deg	0.219*	0.092*	0.199	0.240*	0.097*	0.215
Centrality	0.222*	0.106‡	0.188†	0.242*	0.111‡	0.204†
Clustering coef	0.211*	0.094*	0.188†	0.231*	0.100*	0.202†
Degree	0.233‡	0.095*	0.197	0.256‡	0.099*	0.213
Ego Comp Size	0.227‡	0.096*	0.201*	0.249‡	0.101*	0.215
HITS	0.225*	0.110‡	0.197	0.248*	0.114‡	0.212
PageRank	0.227‡	0.097*	0.200	0.247*	0.101*	0.216
Two Hop Neigh	0.229‡	0.093*	0.195	0.250‡	0.100*	0.212
Static 0.5	0.186	0.077	0.189	0.205	0.081	0.206
Best static	0.218	0.091	0.199	0.239	0.096	0.215

FW – Theoretical background

- We need a theoretical background
  - Why do some predictors work better?

Predictor	CBF	IB	TF-L1	TF-L2	UB	**Median**	**Mean**
ItemSimple	0.257	0.146	0.521	0.564	0.491	0.491	0.396
ItemUser	0.252	0.188	0.534	0.531	0.483	0.483	0.398
RatUser	0.234	0.182	0.507	0.516	0.469	0.469	0.382
RatItem	0.191	0.184	0.442	0.426	0.395	0.395	0.328
IRUser	0.171	-0.092	0.253	0.399	0.257	0.253	0.198
IRItem	0.218	0.152	0.453	0.416	0.372	0.372	0.322
IRUserItem	0.265	0.105	0.523	0.545	0.444	0.444	0.376

Thank you!

Predicting Performance in Recommender Systems

Alejandro Bellogín
Supervied by Pablo Castells and Iván Cantador
Escuela Politécnica Superior
Universidad Autónoma de Madrid

@abellogin
alejandro.bellogin@uam.es

Acknowledgements to the National Science Foundation

for the funding to attend the conference.

Reviewer's comments: Confidence

- Other methods to measure self-performance of RS
  - Confidence

- These methods capture the performance of the RS, not user's performance

Reviewer's comments: Neighbor's goodness

- ❑ Neighbor goodness seems to be a little bit ad-hoc

- We need a measurable definition of neighbor performance

NG(u) ~ "total MAE reduction by u" ~ "MAE without u" – "MAE with u"

$$= \frac{1}{|R_{U-\{u\}}|} \sum_{v \in U-\{u\}} CE_{U-\{u\}}(v) - \frac{1}{|R_{U-\{u\}}|} \sum_{v \in U-\{u\}} CE_{U}(v)$$

$$CE_X(v) = \sum_{i:rat(v,i) \neq \emptyset} |\tilde{r}_X(v,i) - r(v,i)|$$

- Some attempts in trust research: sign and error deviation [Rafter et al. 2009]

Reviewer's comments: Neighbor's weighting issues

- Neighbor size vs dynamic neighborhood weighting
  - So far, only dynamic weighting
    - Same training time than static weighting
  - Future work: dynamic size

- Apply this method for larger datasets
  - Current work

- Apply this method for other CF methods (e.g., latent factor models, SVD)
  - More difficult to identify the combination
  - Future work

Reviewer's comments: Dynamic hybrid issues

- Other methods to combine recommenders
  - Stacking
  - Multi-linear weighting

- We focus on linear weighted hybrid recommendation
- Future work: cascade, stacking