Hungarian Academy of Sciences
Institute for Computer Science and Control

Recommendation systems on Nomao data

Bálint Daróczy
MTA SZTAKI, Ilab
joint work with András Benczúr and Róbert Pálovics

Rating vs. binary matrix
Sparsity: <1% of known values
Overfitting

Singular Value Decomposition

dense representation
no regularization

Stochastic Gradient Descent

sparse representation
vs. conjugate (ALS)
regularization

Both optimize RMSE

Evaluation?

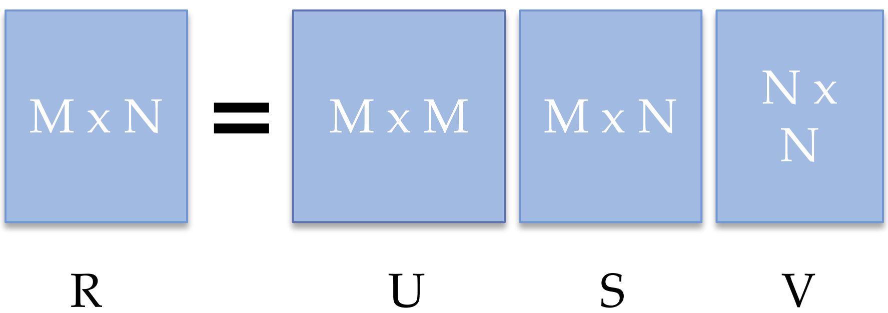
User/Movie	Napoleon Dynamite	Monster Inc.	Cindarella	Life on Earth
David	1	?	?	3
Dora	5	3	5	5
Peter	?	4	3	?

User/Movie	Napoleon Dynamite	Monster Inc.	Cindarella	Life on Earth
David	1	0	0	1
Dora	1	1	1	1
Peter	0	1	1	0

User/Movie	Napoleon Dynamite	Monster Inc.	Cindarella	Life on Earth
David	1			1
Dora	1	1	1	1
Peter		1	1	

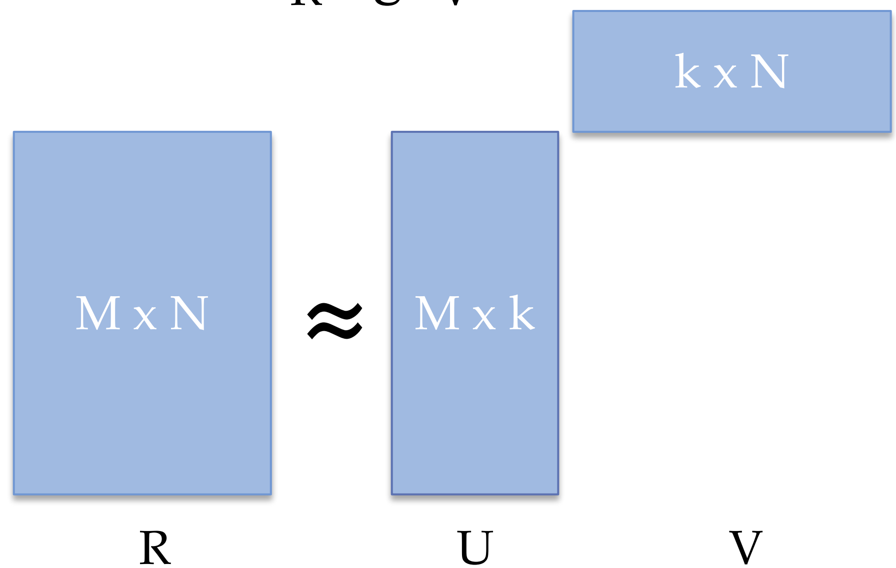
Singular Value Decomposition

$$R = U^T S V$$



Stochastic Gradient Descent

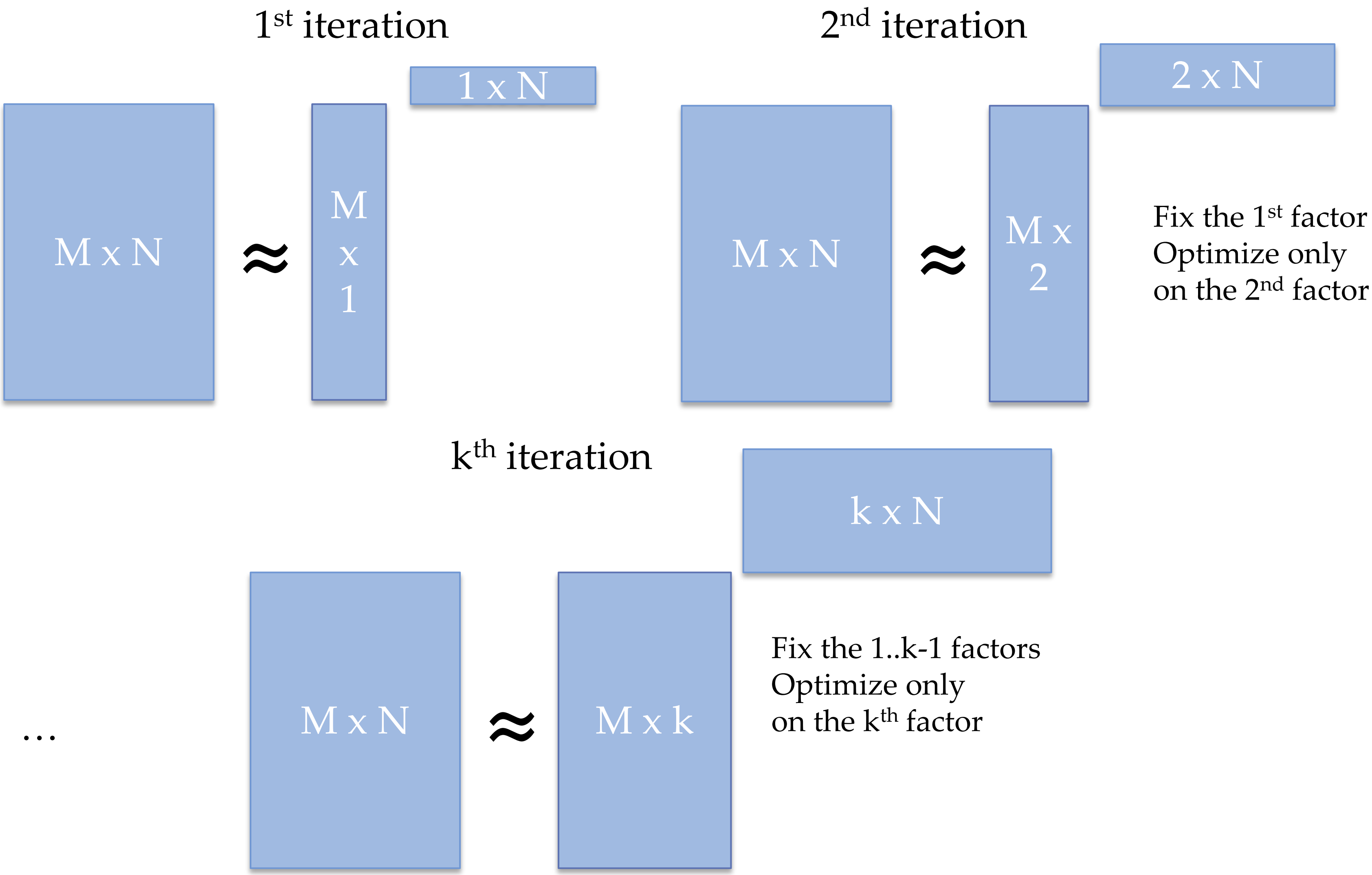
$$R = U^T V$$



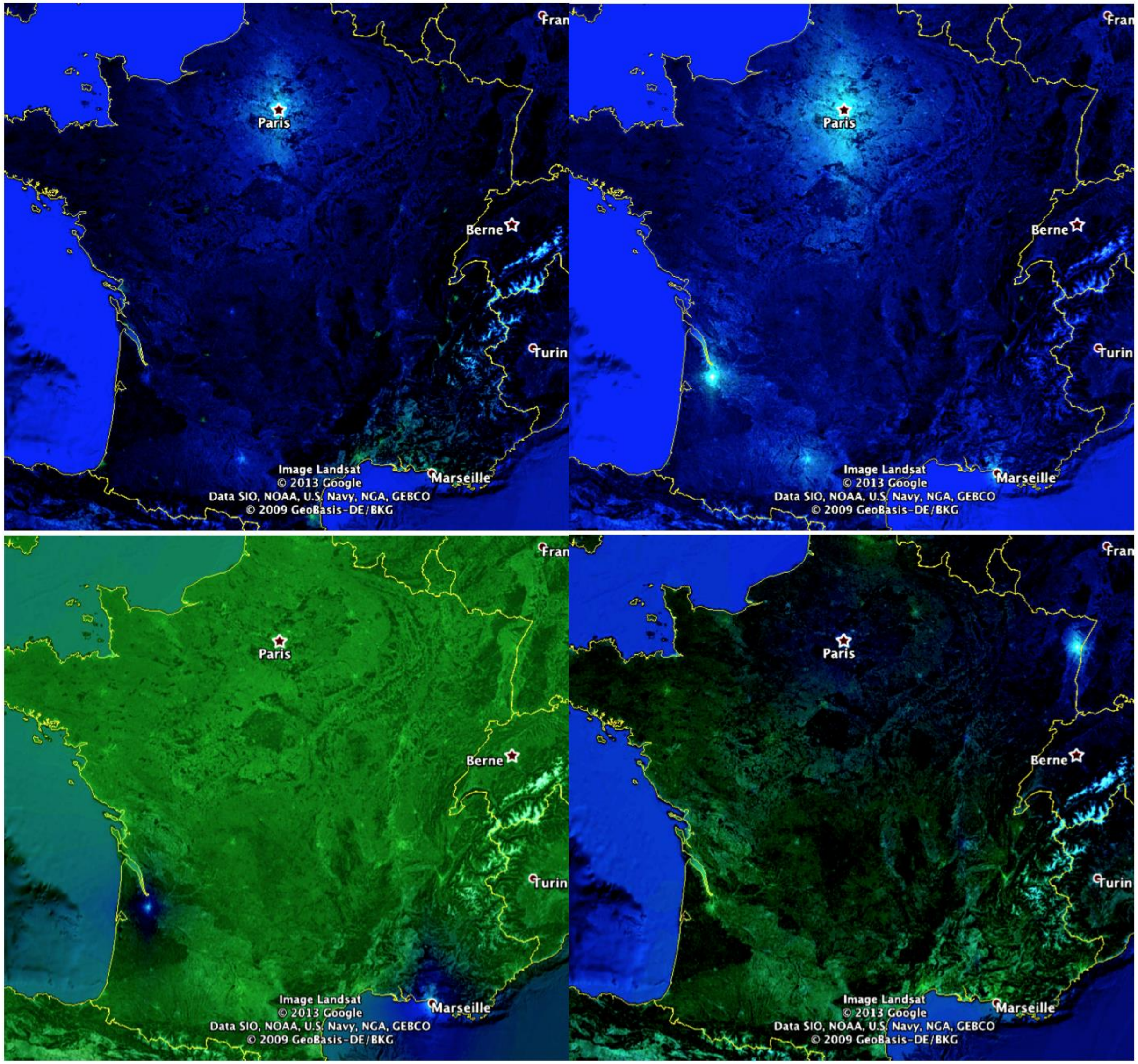
In our case:
M: number of users
N: number of items
R: the original (<1% known) rating matrix

In comparison to SVD, the SGD factors are not ranked
Ranked factors: iterative SGD optimize only on a single factor at a time

Iterative Stochastic Gradient Descent (by Simon Funk)



Singular Value Decomposition

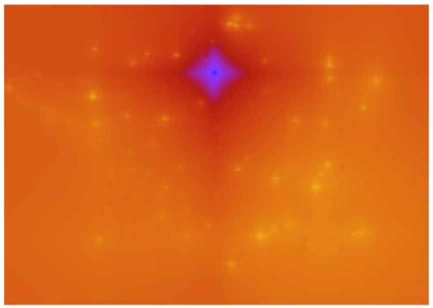
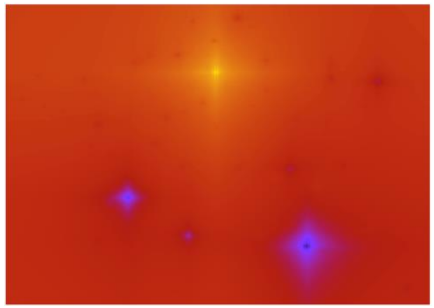
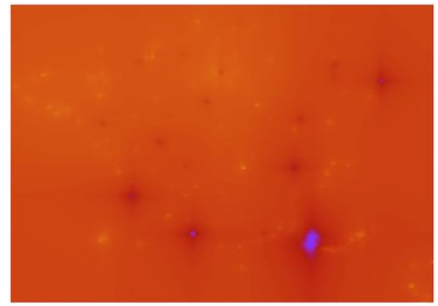
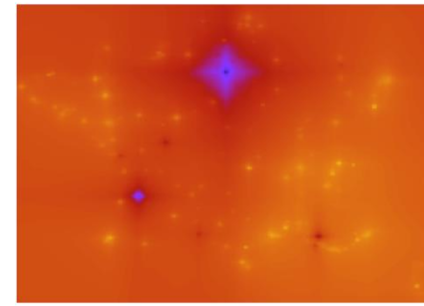
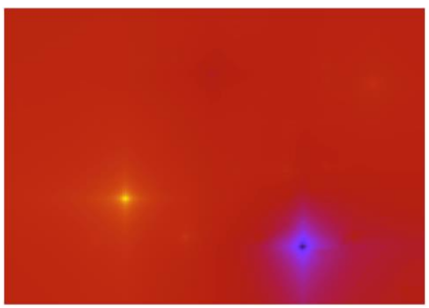
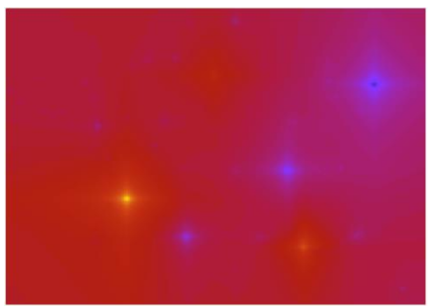
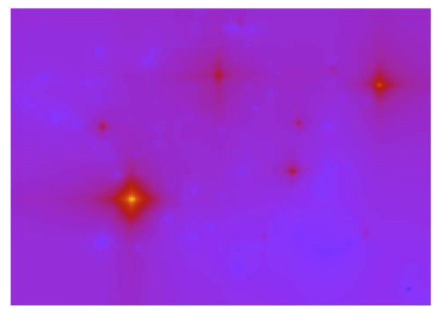
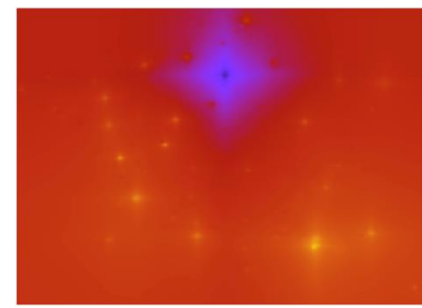
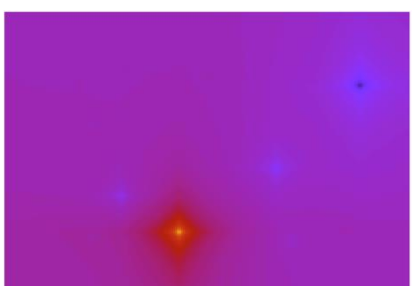


The first 4 factors mapped over France

The first 5 factors mapped over France

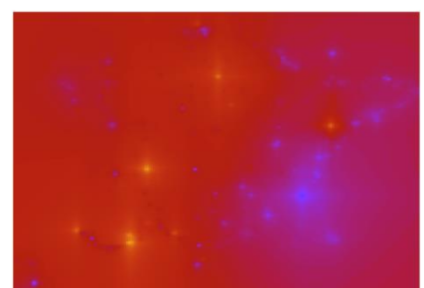
Singular Value Decomposition

Stochastic Gradient Descent not ranked!

Smoothness

képlet



Recommend locations near to already visited places
vs.
Expand/modify the training set or regularization

Expansion via locality

- SVD vs. SGD
- Binary vs. Rating matrix
- identifying neighbors: k-nearest vs. radius , travel time?
- number of neighbors (n)?

Let be E the set of known ratings and N_j the neighbors of the location j, than we can modify the training set as follows. For all (u,i)

$$\hat{r}_{u,i} = \begin{cases} r_{u,i} & \text{if } (u,i) \in \mathrm{E} \\ f(R_u, N_{u,i}) & \text{if } (u,i) \notin \mathrm{E} \text{ and } \exists\, j \text{ with } (u,j) \in \mathrm{E} \text{ and } i \in N_j \\ 0 \text{ or don't care} & \text{otherwise} \end{cases}$$

where f is function of R_u, the set of known ratings by user "u" and $N_{u,i}$, the set locations visited by "u"where "i" is a place of their neighorhood.

Model 1: expand the list of locations per user with the neighbors of visited places

a) learn the ratings

$$f(R_u, N_{u,i}) = \frac{1}{|N_{u,i}|} \sum_{j \in N_{u,i}} r_{u,j}$$

or a constant

$$f(R_u, N_{u,i}) = c$$

b) learn the occurrence

$$f(R_u, N_{u,i}) = 1$$

Model 2: adaptive distance based expansion, smoothed with local density

a) learn the ratings

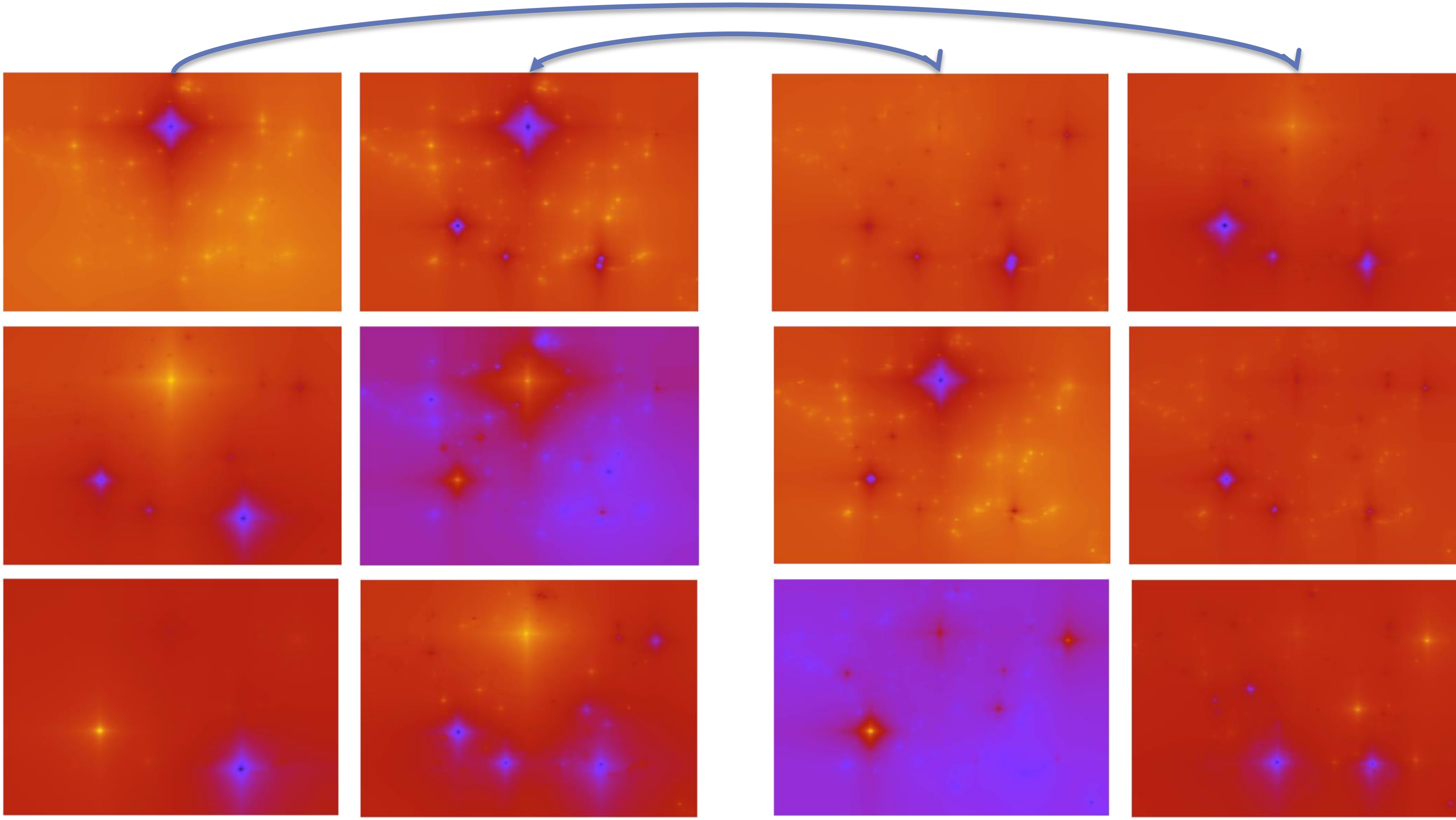
$$f(R_u, N_{u,i}) = \frac{1}{|N_{u,i}|} \sum_{j \in N_{u,i}} \hat{r}_{u,j} \mathrm{e}^{-\frac{d_{L2}(i,j)}{\hat{d}_{L2}(j)}}$$

b) learn the occurrence

$$f(R_u, N_{u,i}) = \mathrm{e}^{-\frac{d_{L2}(i,j)}{\hat{d}_{L2}(j)}}$$

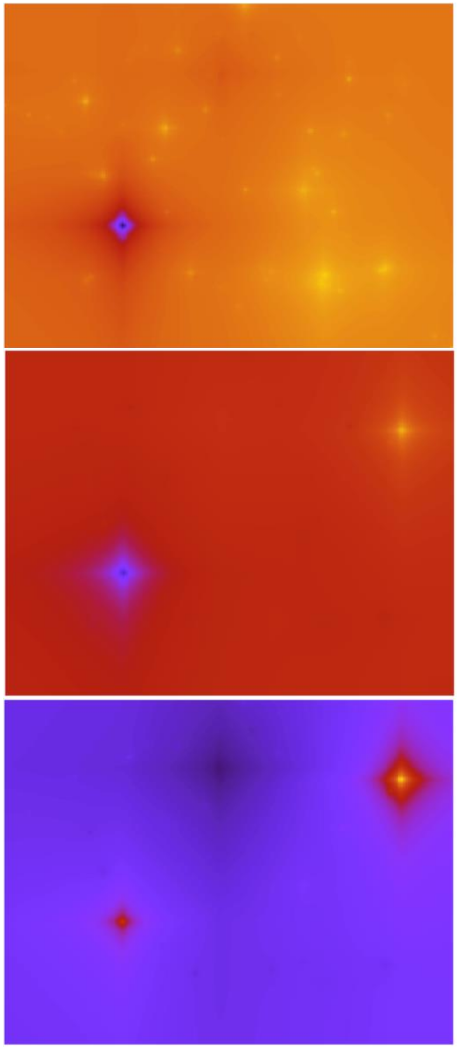
Effect of “n”, The first 5 factors mapped over France

Singular Value Decomposition | Stochastic Gradient Descent



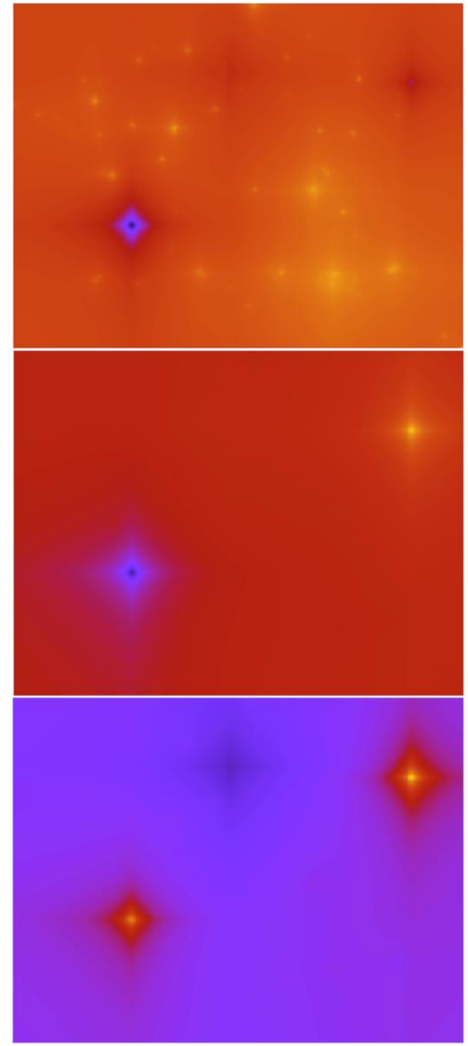
original | n=1 | original | n=1

Effect of "n"



n=1

≈



n=20

Performance measures

RMSE: $RMSE = \sqrt{\sum_{(u,i)} (r_{u,i} - r^*_{u,i})^2}$

$$RMSE_{sparse} = \sqrt{\sum_{(u,i)\in E} (r_{u,i} - r^*_{u,i})^2)}$$

Recall @ K: number of hits/number of relevant items

$$Recall(K) = \frac{1}{|U|} \sum_u Recall_u(K)$$

per user

$$Recall_u(K) = \frac{1}{|R_u|} \sum_{i=1}^{K} rel_{u,i}$$

Normalized Discounted Cumulative Gain @ K

$$nDCG(K) = \frac{1}{|U|} \sum_u nDCG_u(K)$$

per user

$$nDCG_u(K) = \frac{DCG_u(K)}{iDCG_u(K)} \quad \text{where} \quad DCG_u(K) = rel_{u,1} + \sum_{i=2}^{K} \frac{rel_{u,i}}{log_2(i)}$$

Item	Rank for a user	Relevance to the user
item1	0	0
item2	1	1
…	…	0
		1
		0
		0
		1
item K-1	K-2	0
item K	K-1	1

Relevance ($rel_{u,i}$)?

Binary or real

Preliminary results

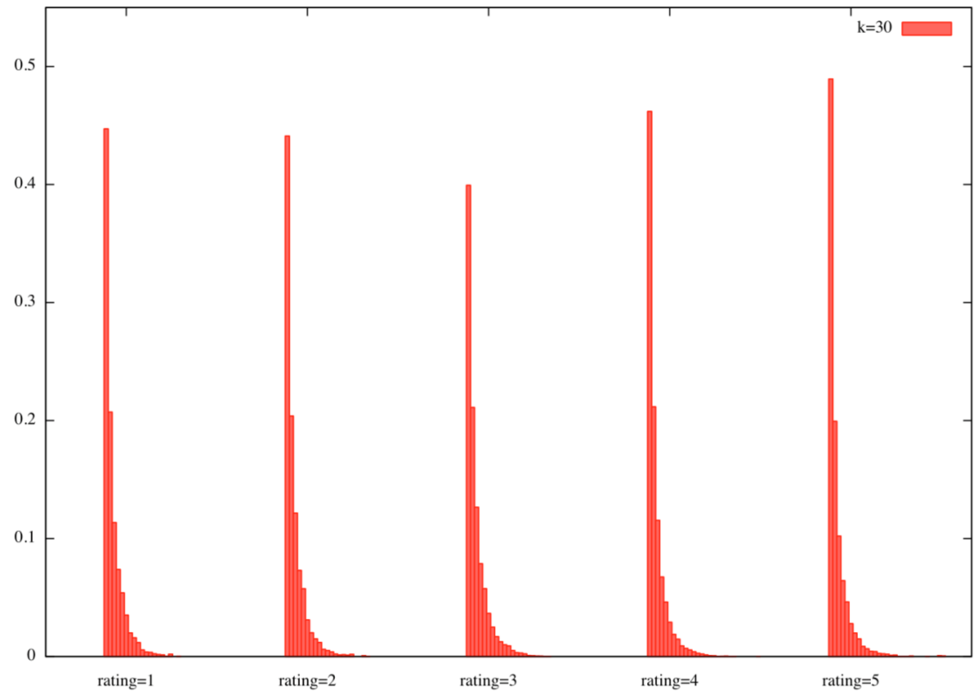
Datasets

Nomao: France, mostly Paris
7605 location
9471 users
97453 known ratings

Yelp: U.S.A
45981 users
11537 locations
227906 known ratings

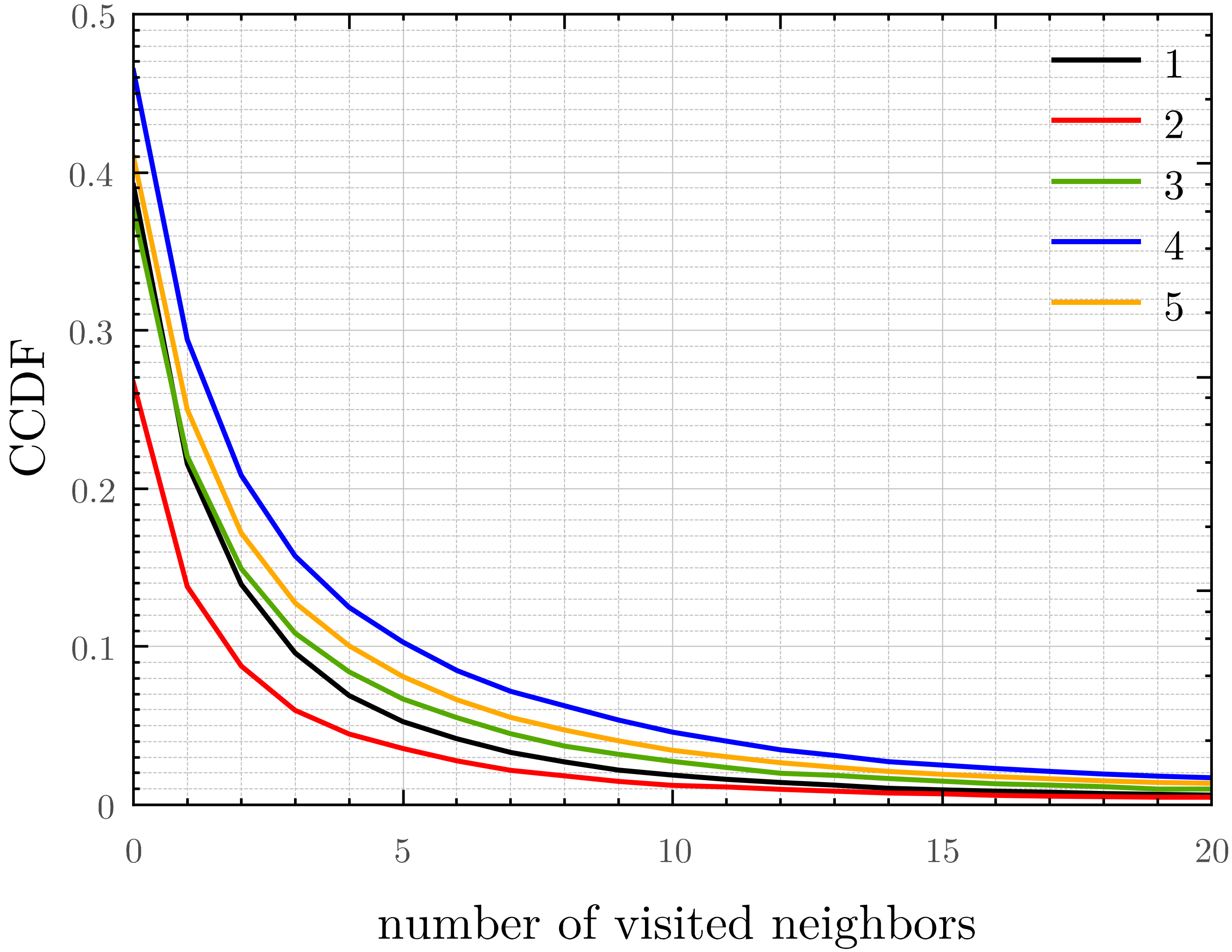
"Rating effect"

For a given user the neighbors of "average" rated places are more-likely visited as the neighbors of "extremely" rated places

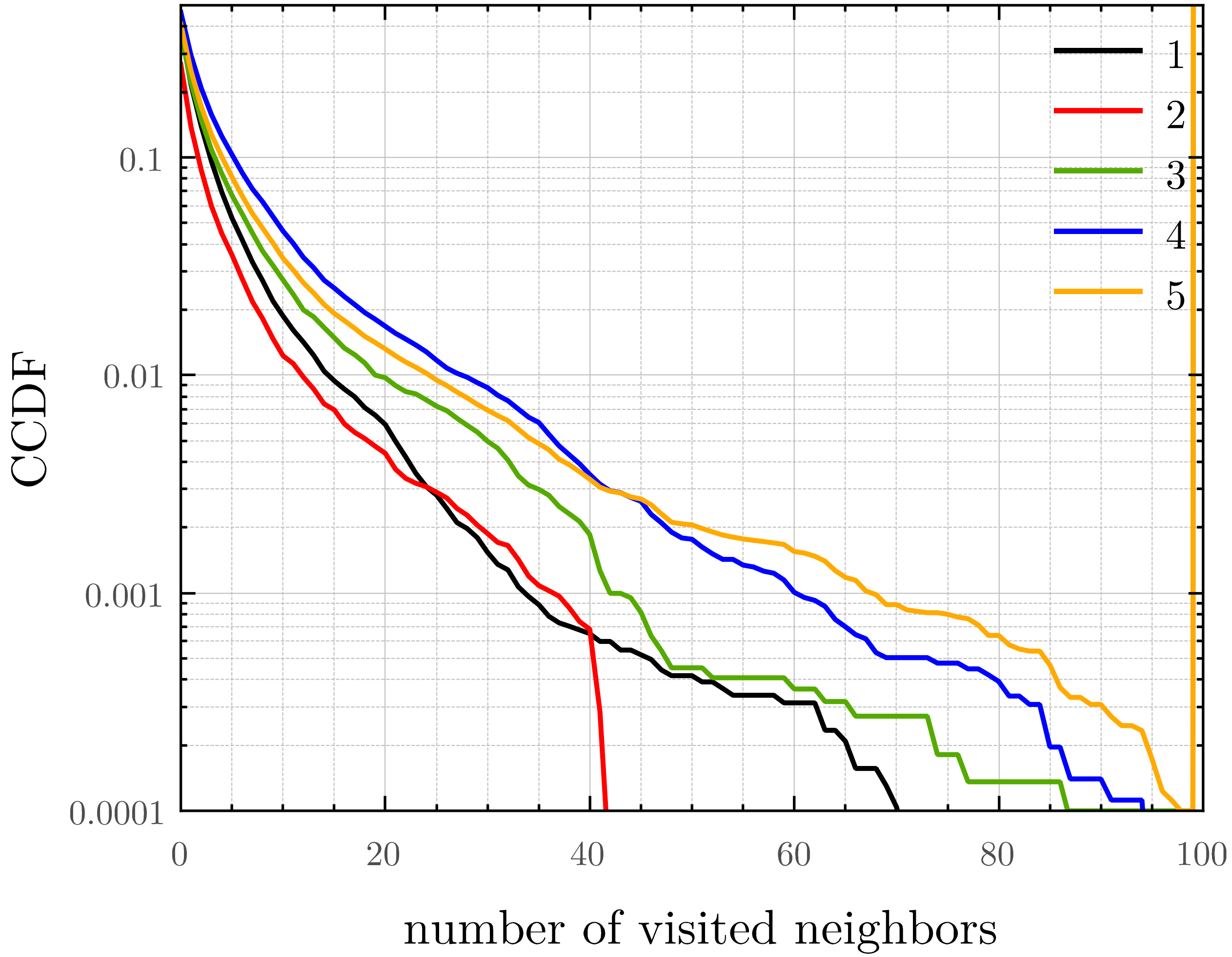


Refine recommendation: regularization or re-ranking
Location adaptive expansion via the ratings of the visited places

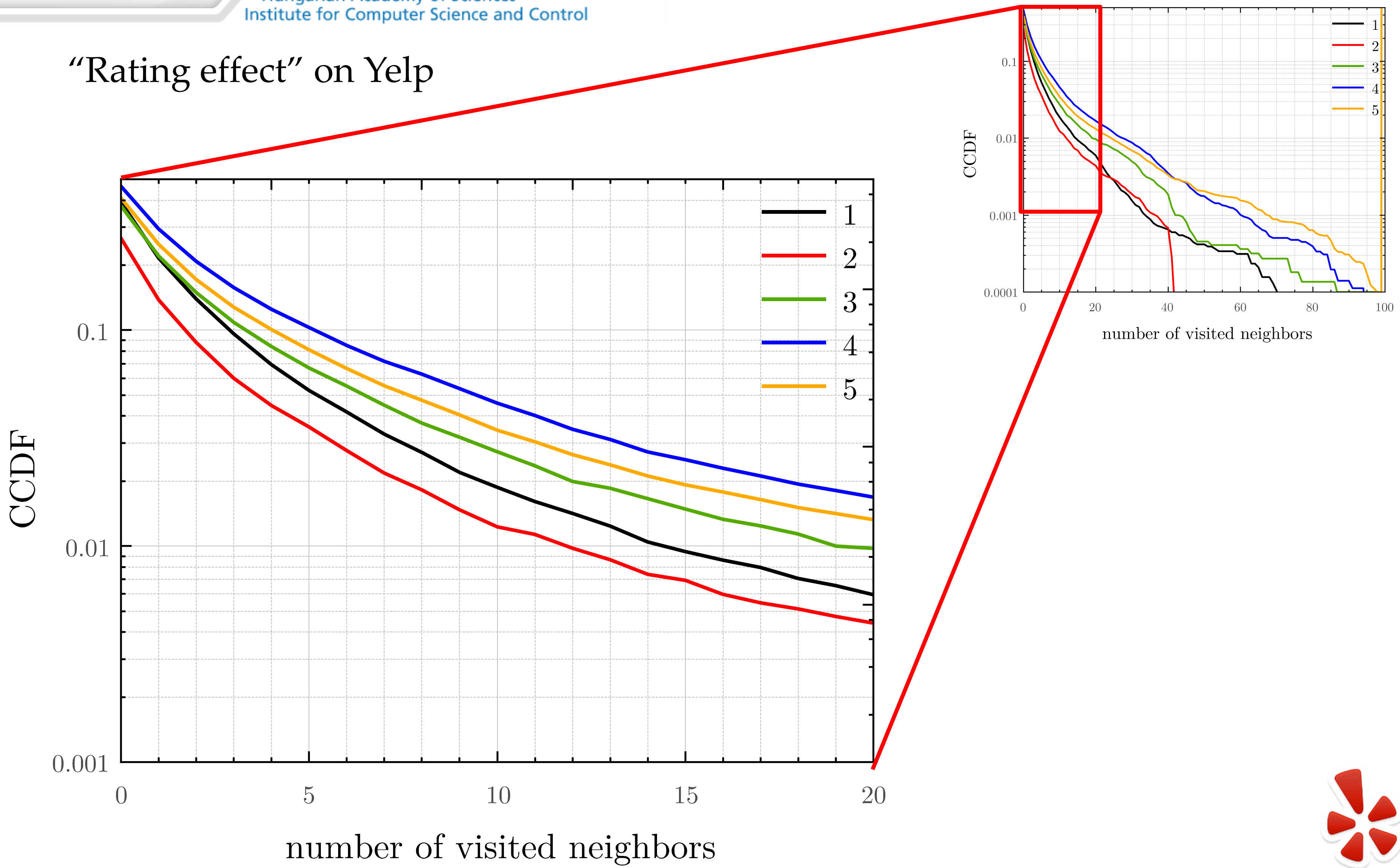
"Rating effect" on Yelp



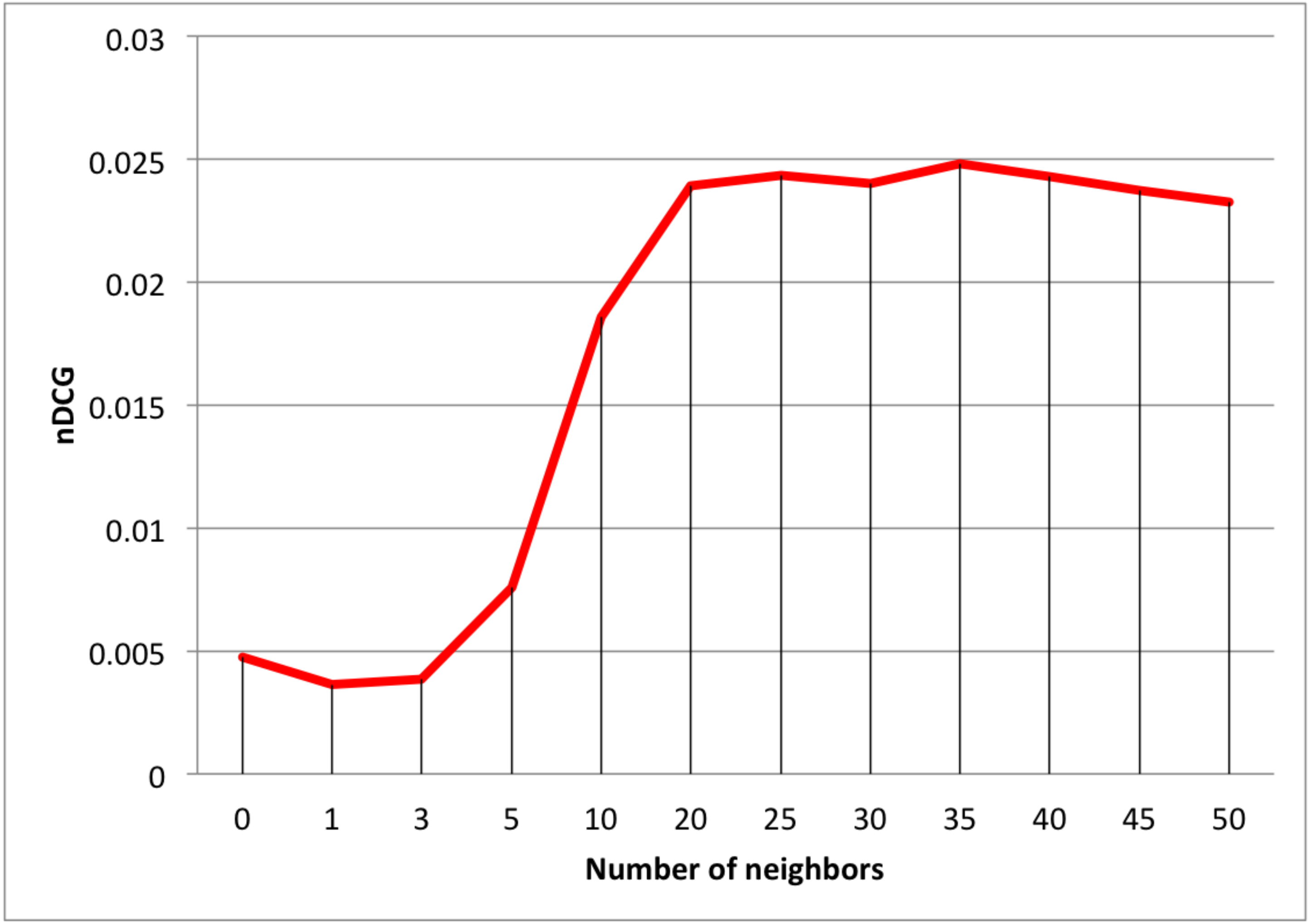
"Rating effect" on Yelp (log-scale)



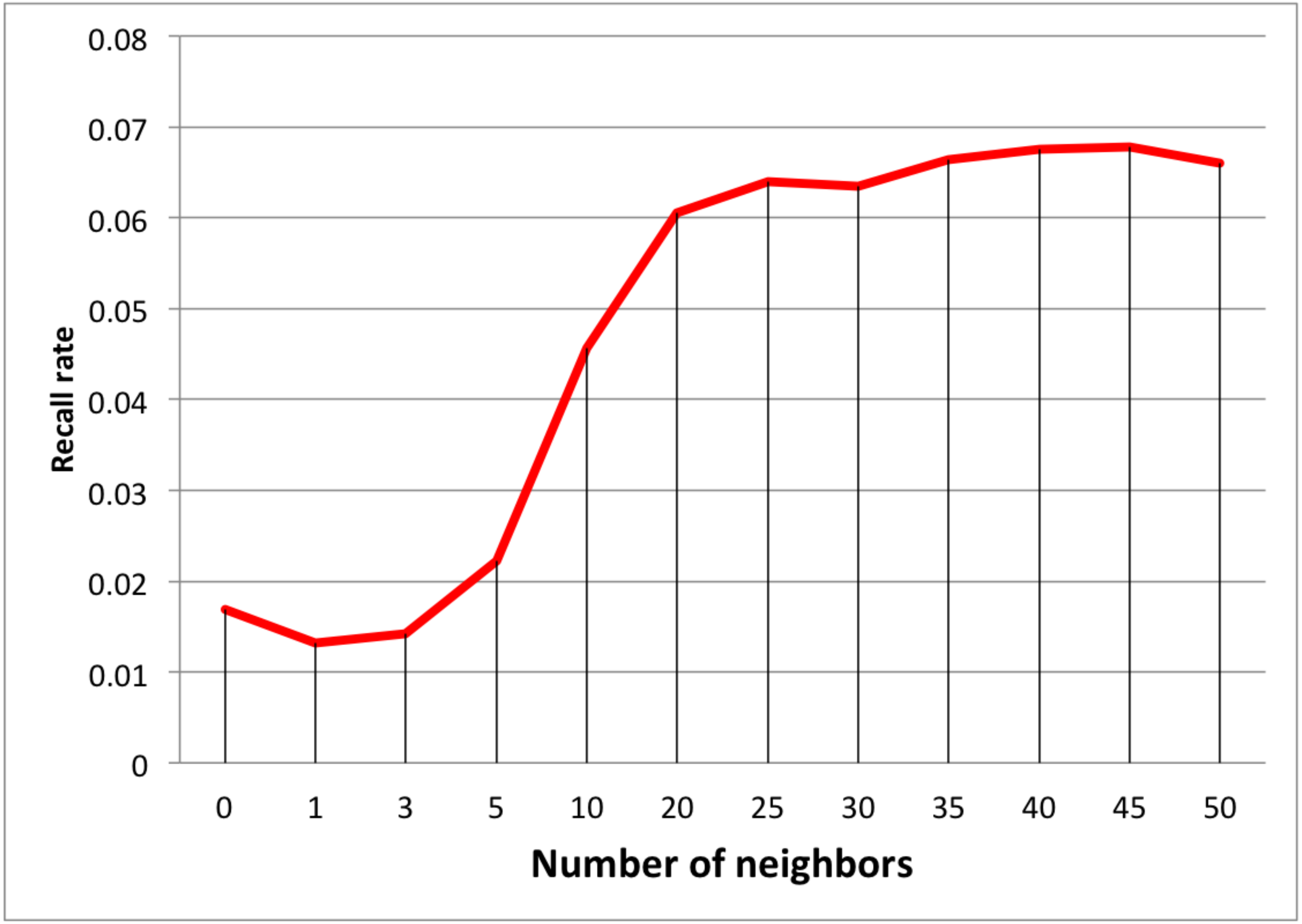
“Rating effect” on Yelp



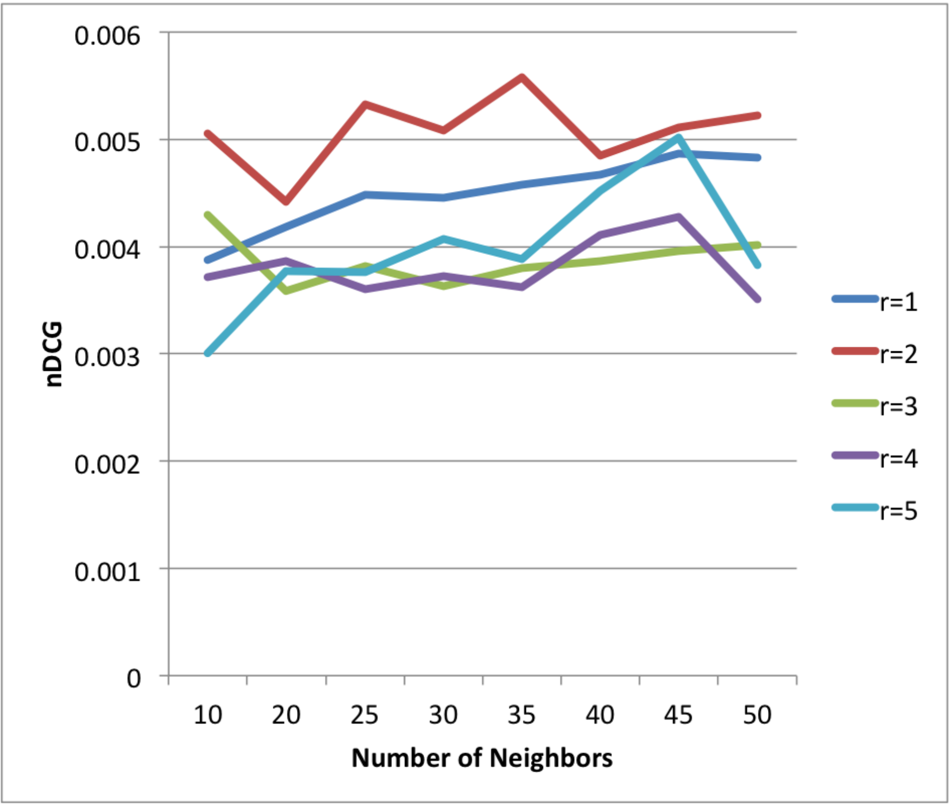
M1a: Expansion with the original ratings (nDCG@100)

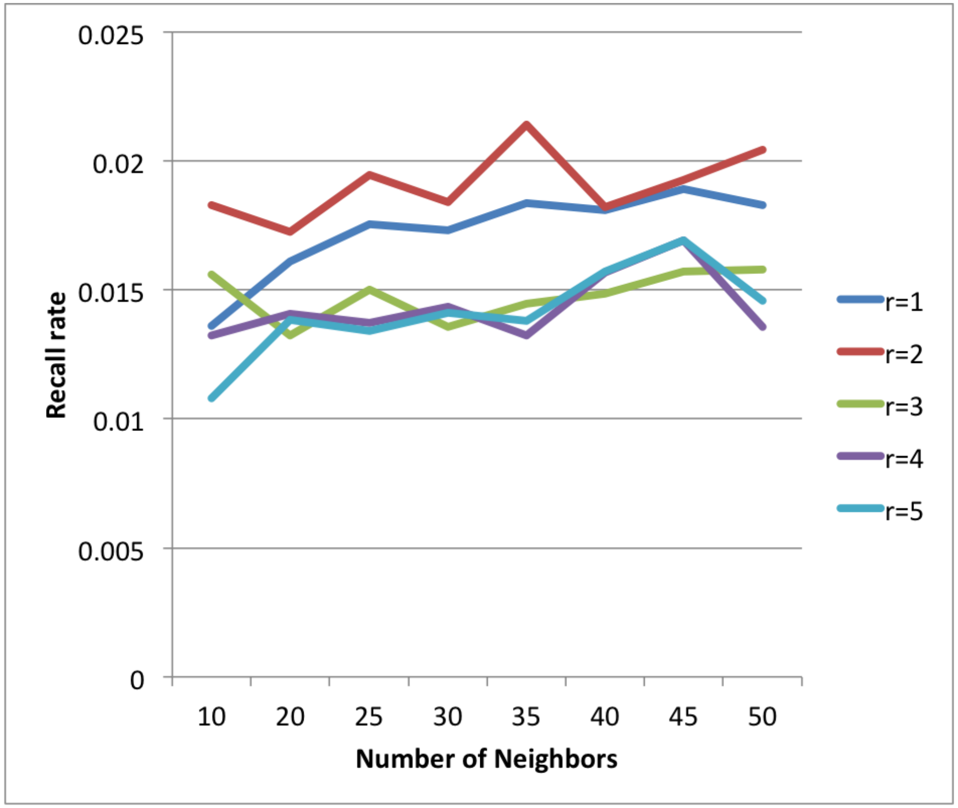


M1a: Expansion with the original ratings (Recall@100)

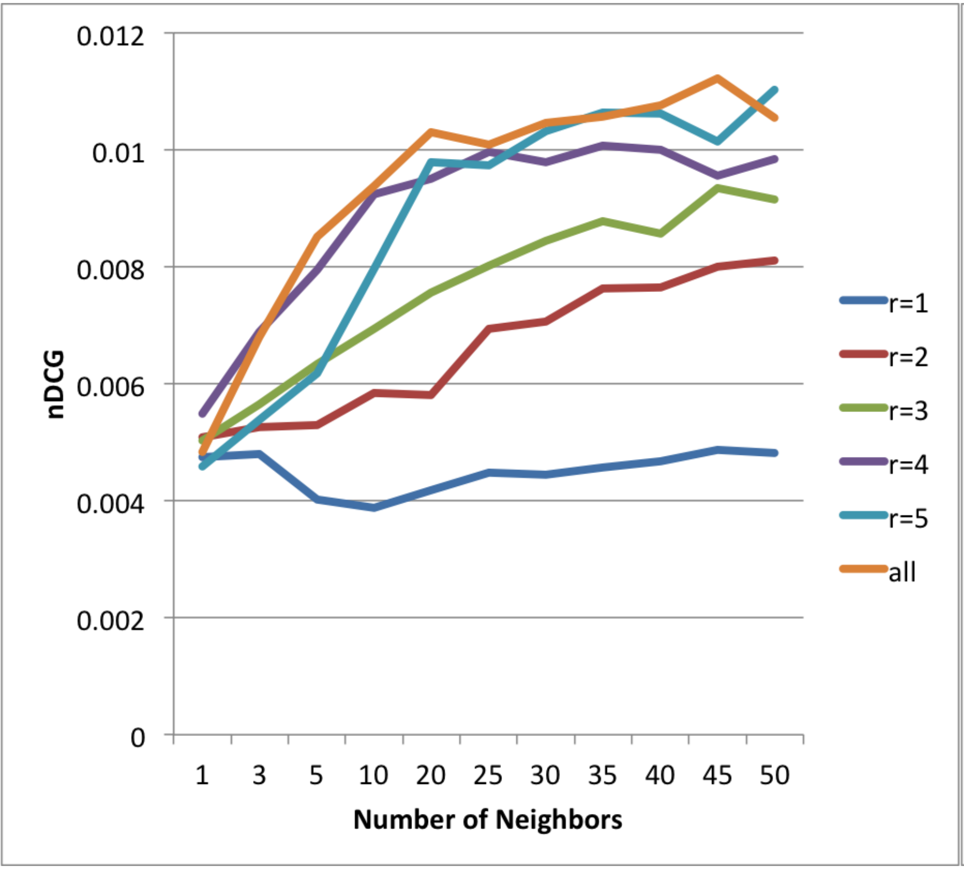


M1a: Weighted expansion per rating (nDCG@100 and Recall rate)
Note: we lower the test predicitions if the original rating was “low”

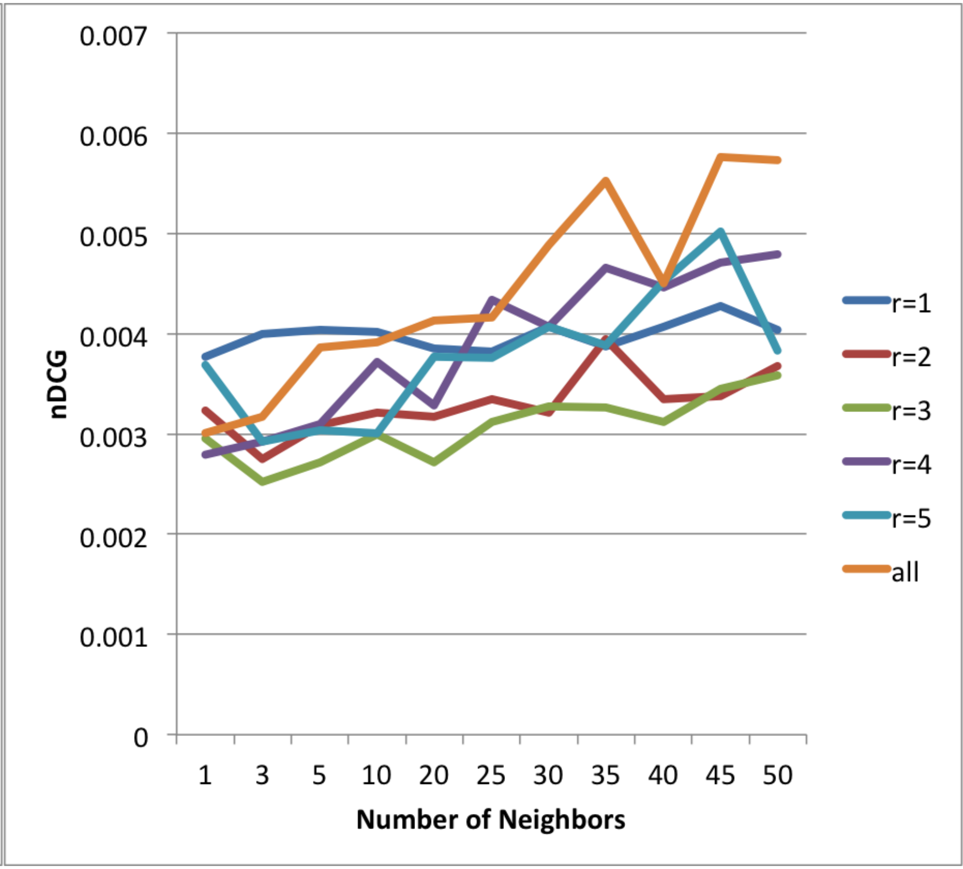




M1a: Constant expansion per rating (nDCG@100)

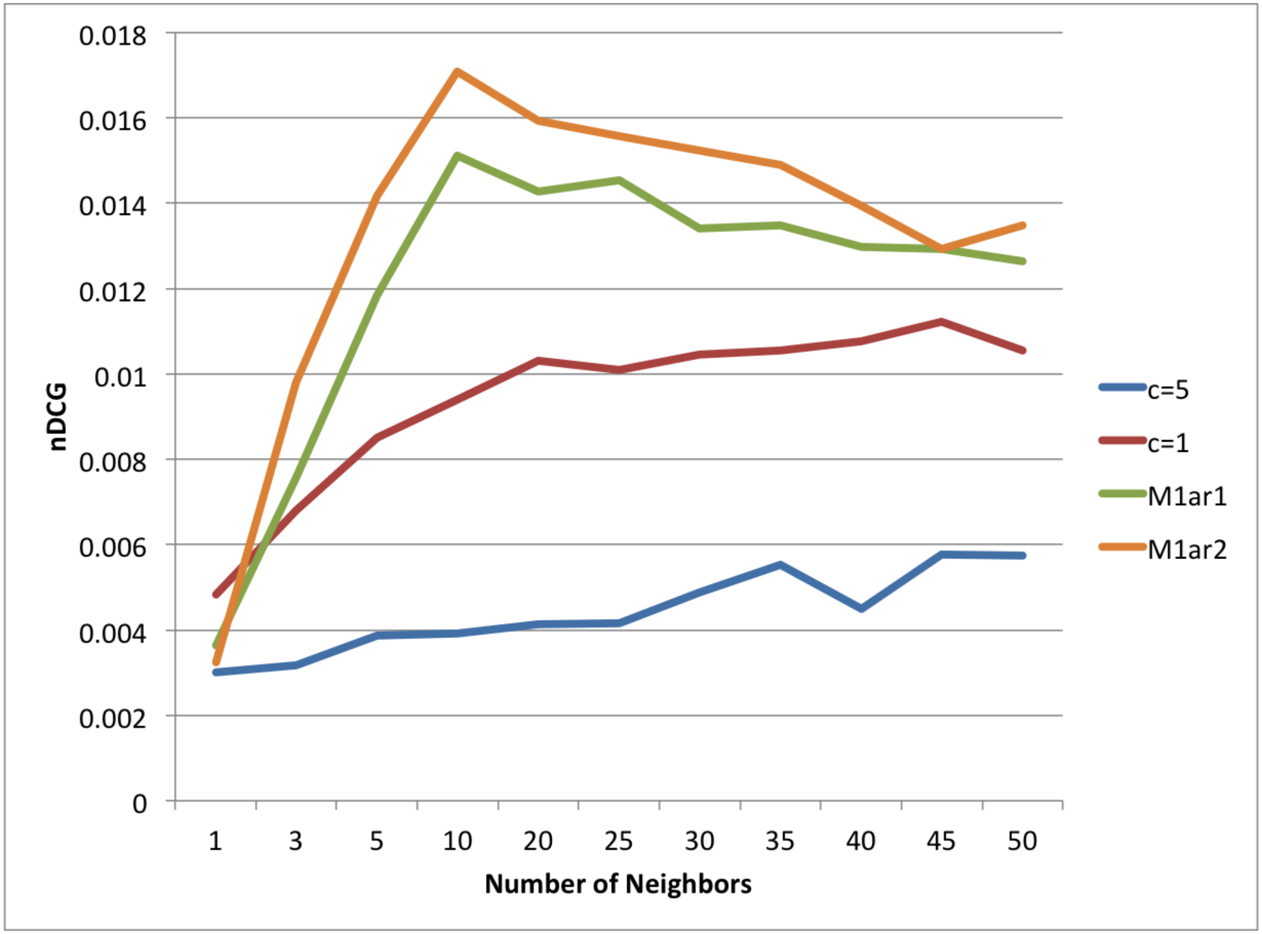


c=1



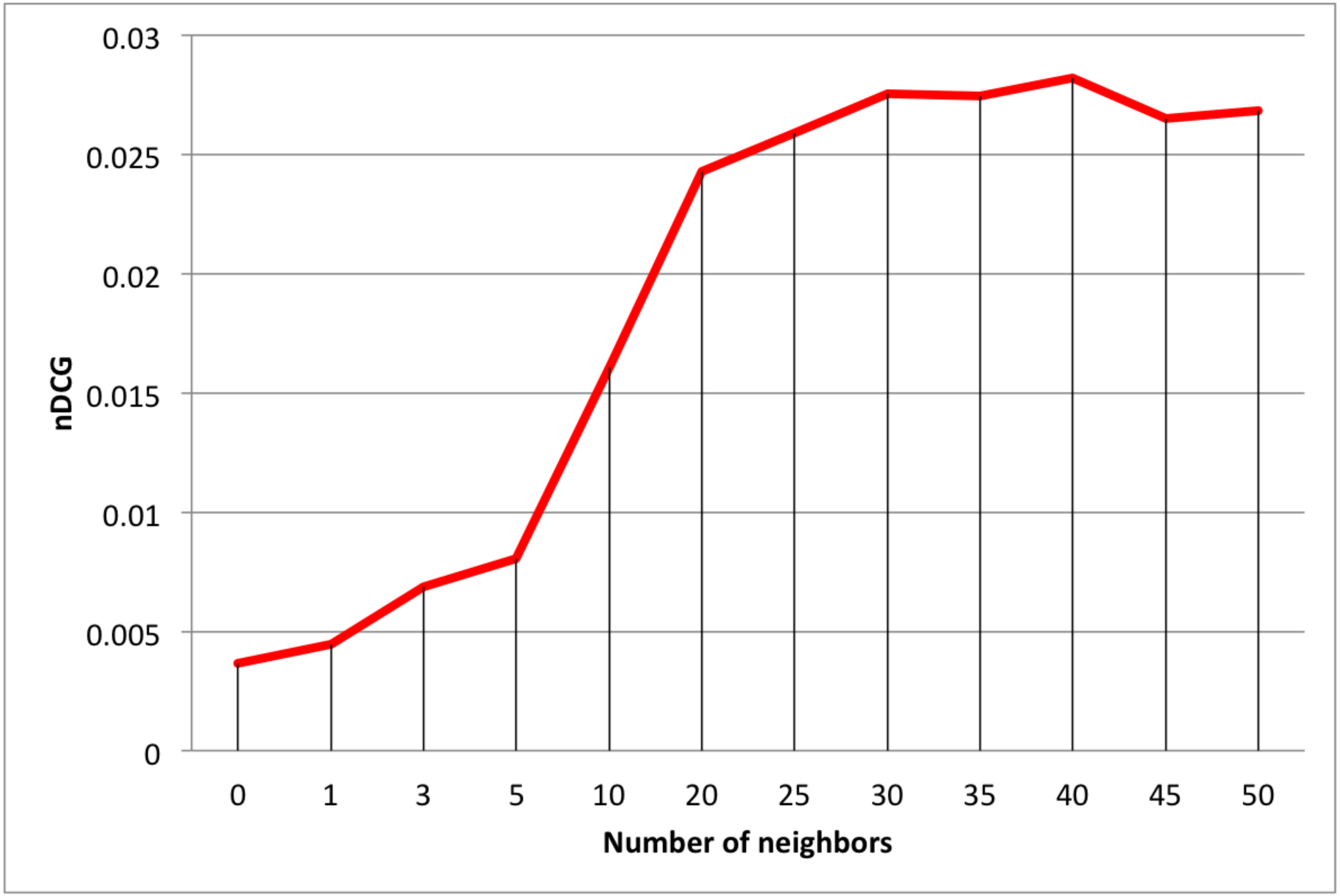
c=5

M1ar: Rating dependent constant expansion per rating (nDCG@100)

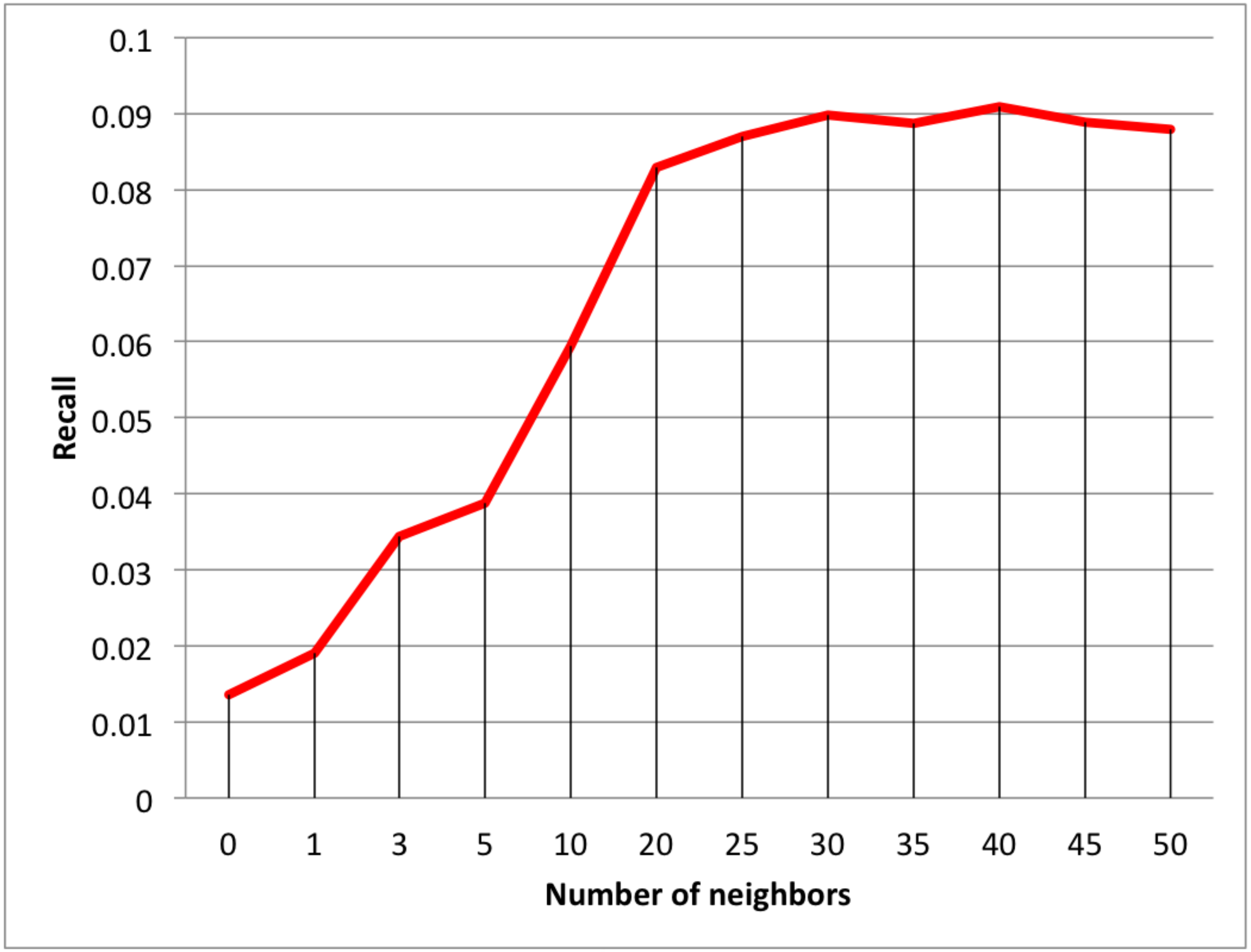


M1ar1: c=1 if r=1,2,3 and c=5 if r=4,5
M1ar2: c=1 if r=1,5 and c=5 if r=2,3,4

M1b: Expansion the list of visited locations with neighbors (nDCG@100)



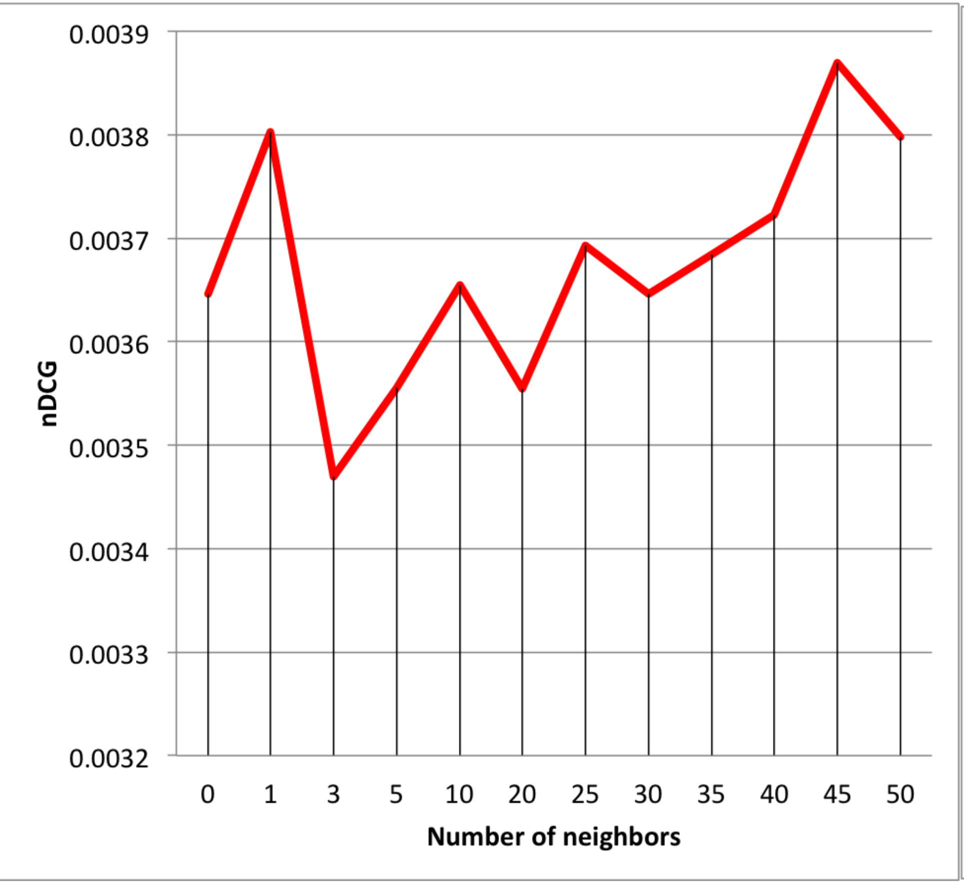
M1b: Expansion the list of visited locations with neighbors (Recall@100)

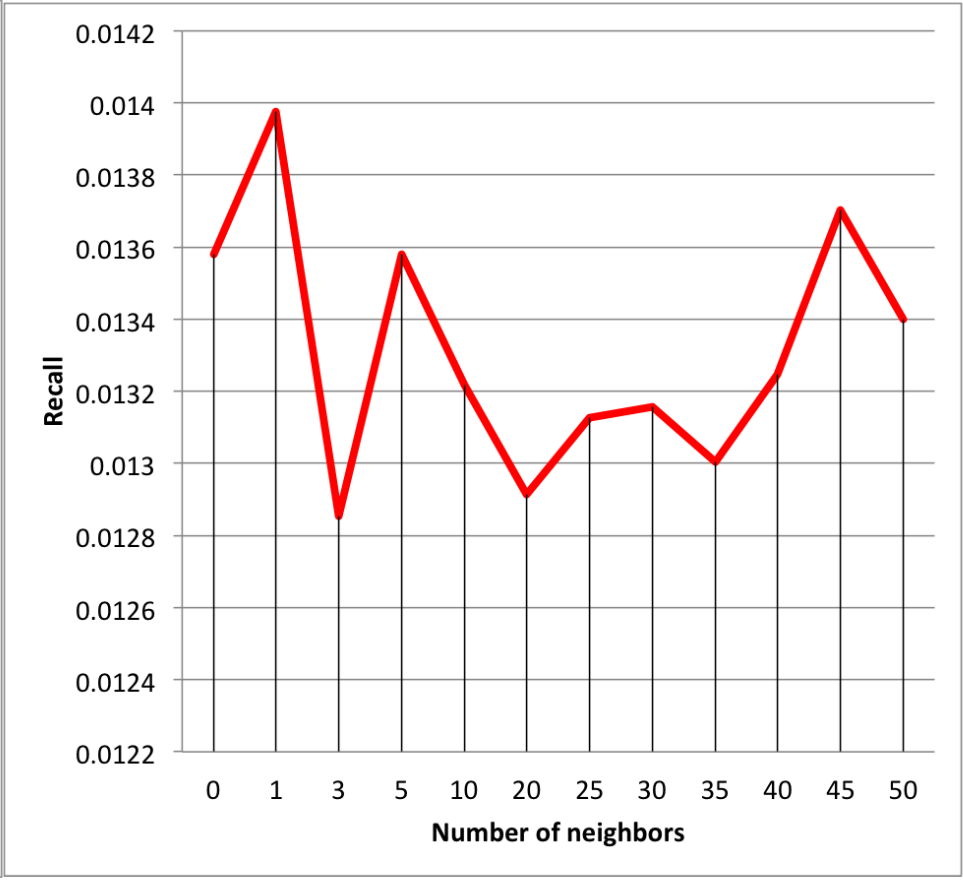


MTA SZTAKI
Hungarian Academy of Sciences
Institute for Computer Science and Control

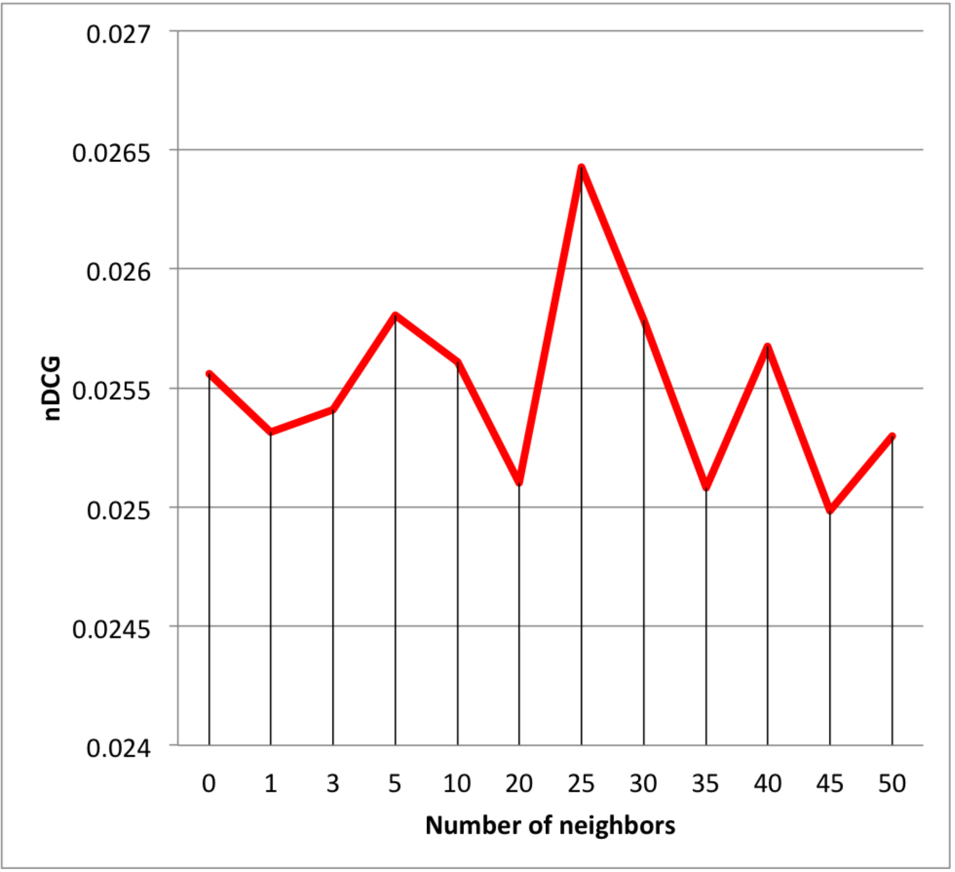
nomad

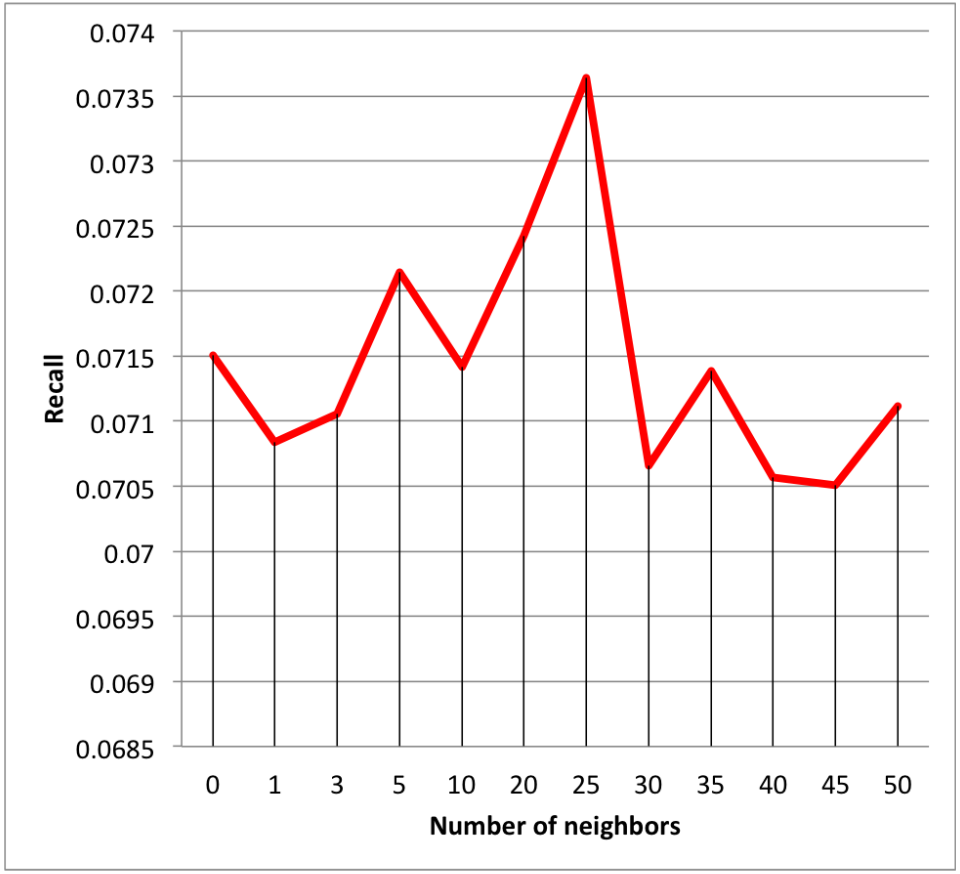
M2b: Distance adaptive expansion of visited locations , smoothed (nDCG@100 and Recall)



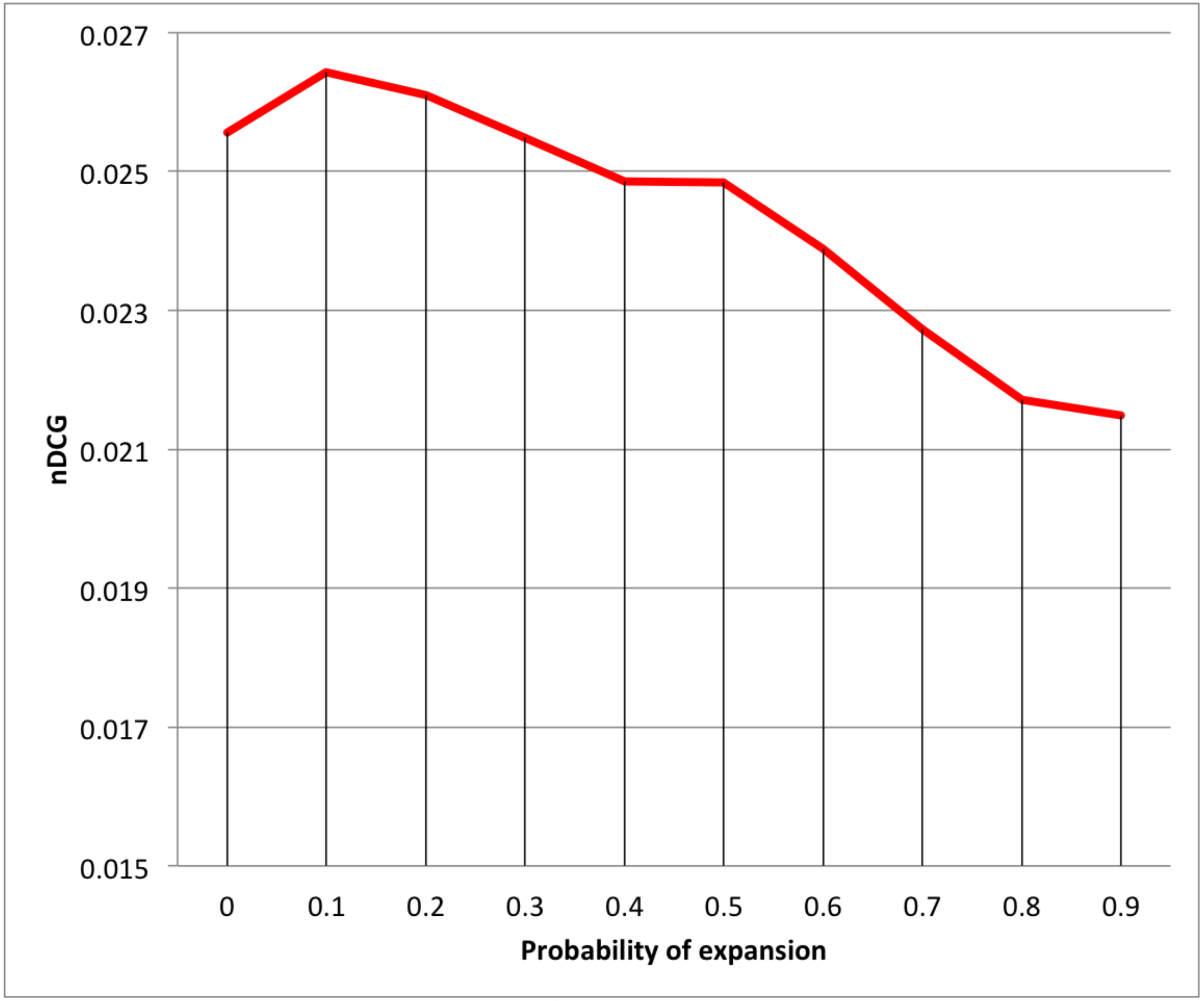


M2a: Distance adaptive expansion of ratings , smoothed (nDCG@100 and Recall)





M2a: Probability of expansion



Conclusions and future work

- SGD and SVD “factors” are similar
  - factors with highest eigenvalue are mostly correlated with a particular place
- “Rating effect”
  - rating dependent distribution of visited neighbors
  - observed over Nomao and Yelp too
- In some cases expansion via neighbors of visited places could increase the performance

Next steps:

- Combination of non-factor and factor models
- We just started to use the “rating effect”: probabilistic models
- MultiMF: Learn where to expand

Thank you! Questions?