CWI

Understanding Similarity Metrics in Neighbour-based Recommender Systems

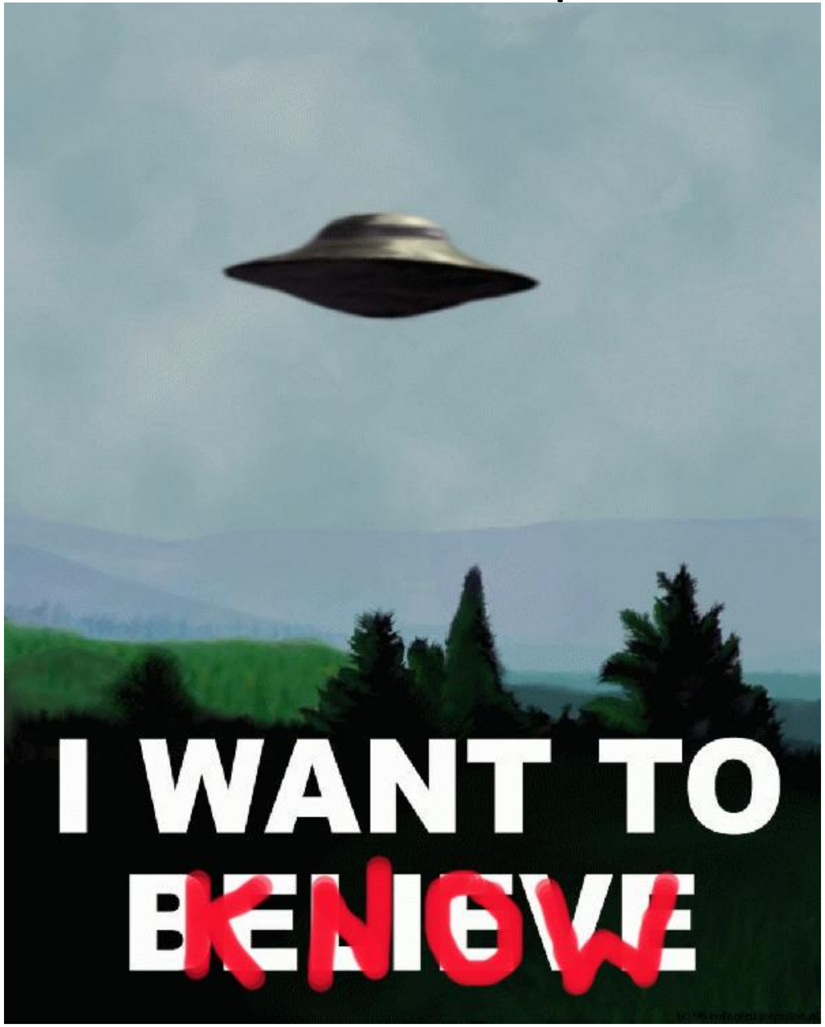
Alejandro Bellogín, Arjen de Vries
Information Access
CWI

ICTIR, October 2013

ERCIM

Motivation

- Why some recommendation methods perform better than others?



Motivation

- Why some recommendation methods perform better than others?

- Focus: nearest-neighbour recommenders
  - What aspects of the similarity functions are more important?
  - How can we exploit that information?

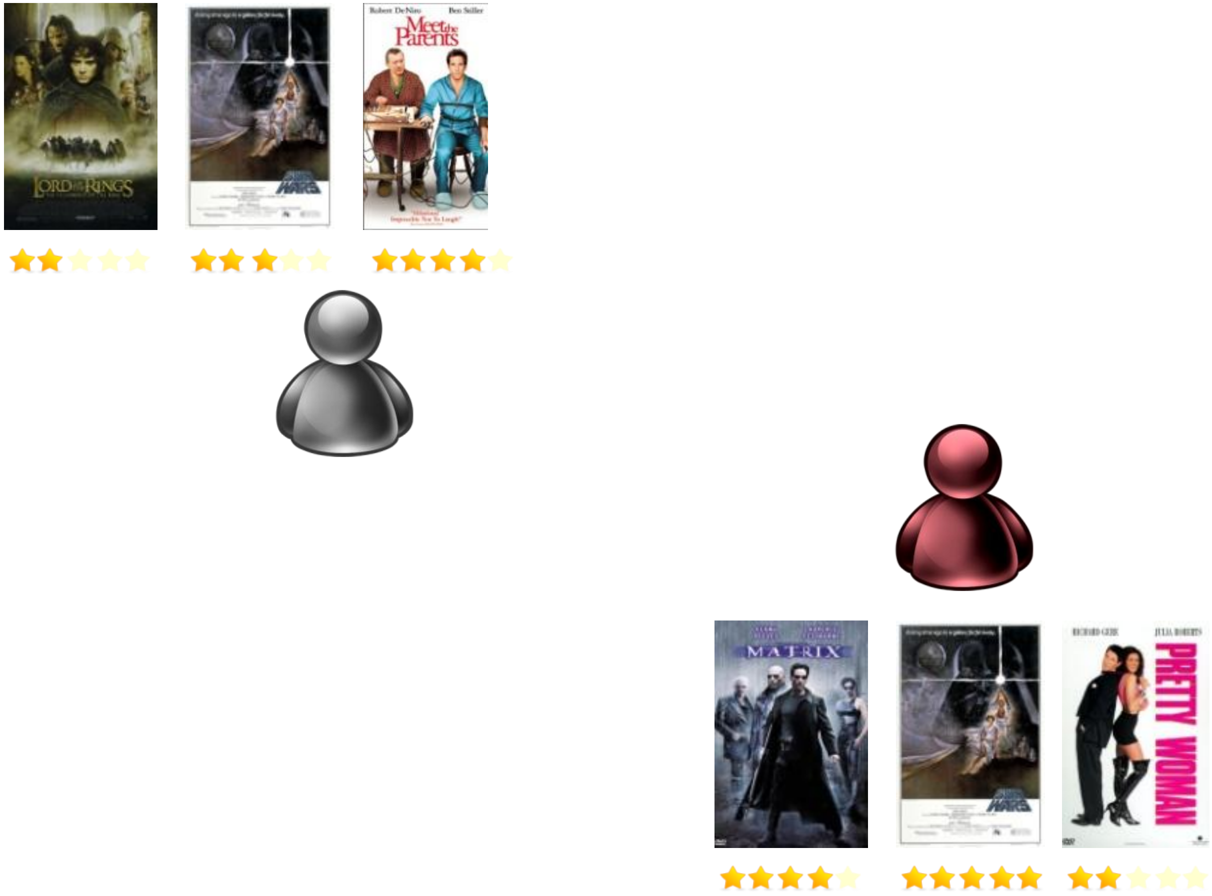
Context

- Recommender systems
  - Users interact (rate, purchase, click) with items

MATRIX
PRETTY WOMAN
★★★★☆
★★★★★
★★☆☆☆

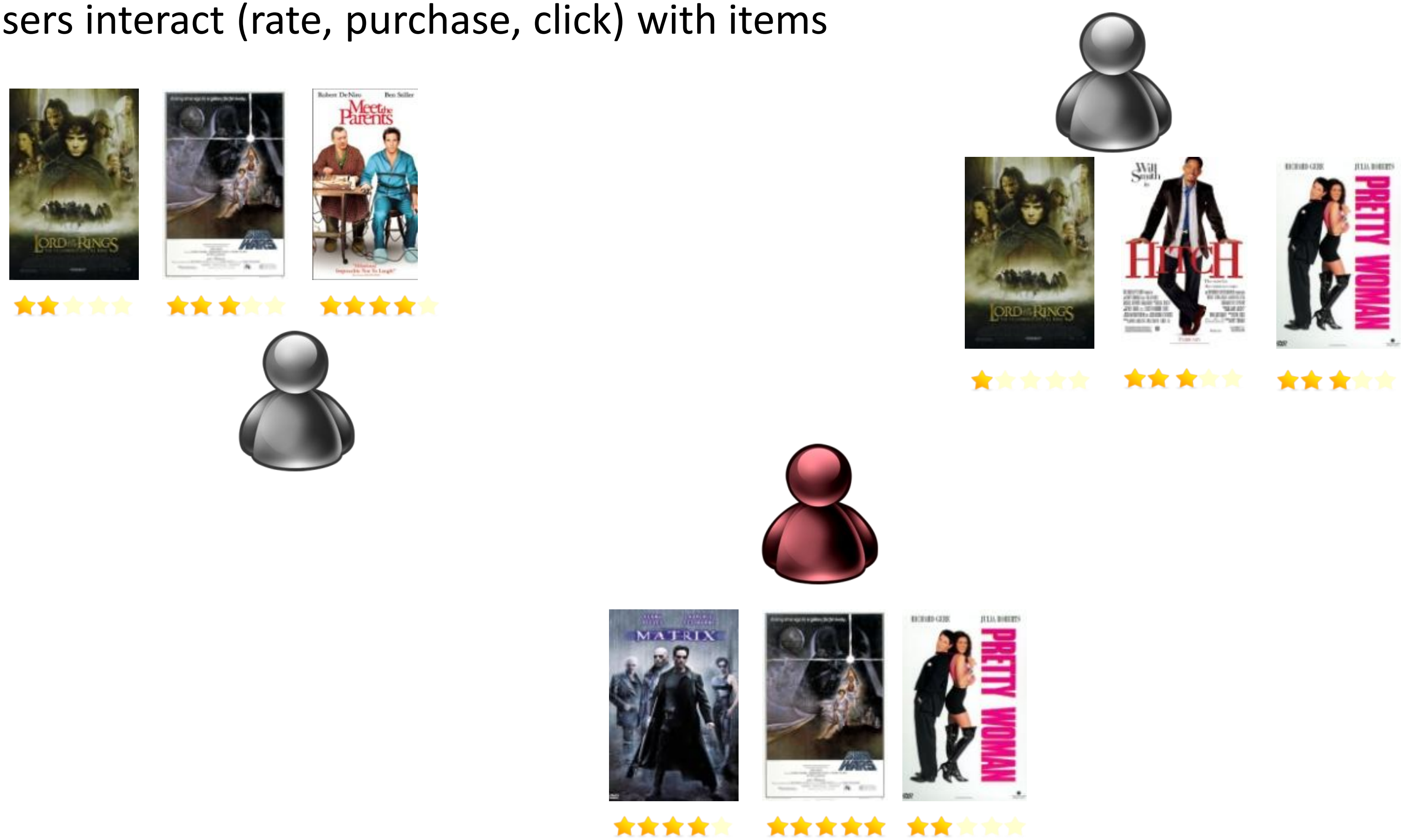
Context

- Recommender systems
  - Users interact (rate, purchase, click) with items



Context

- Recommender systems
  - Users interact (rate, purchase, click) with items



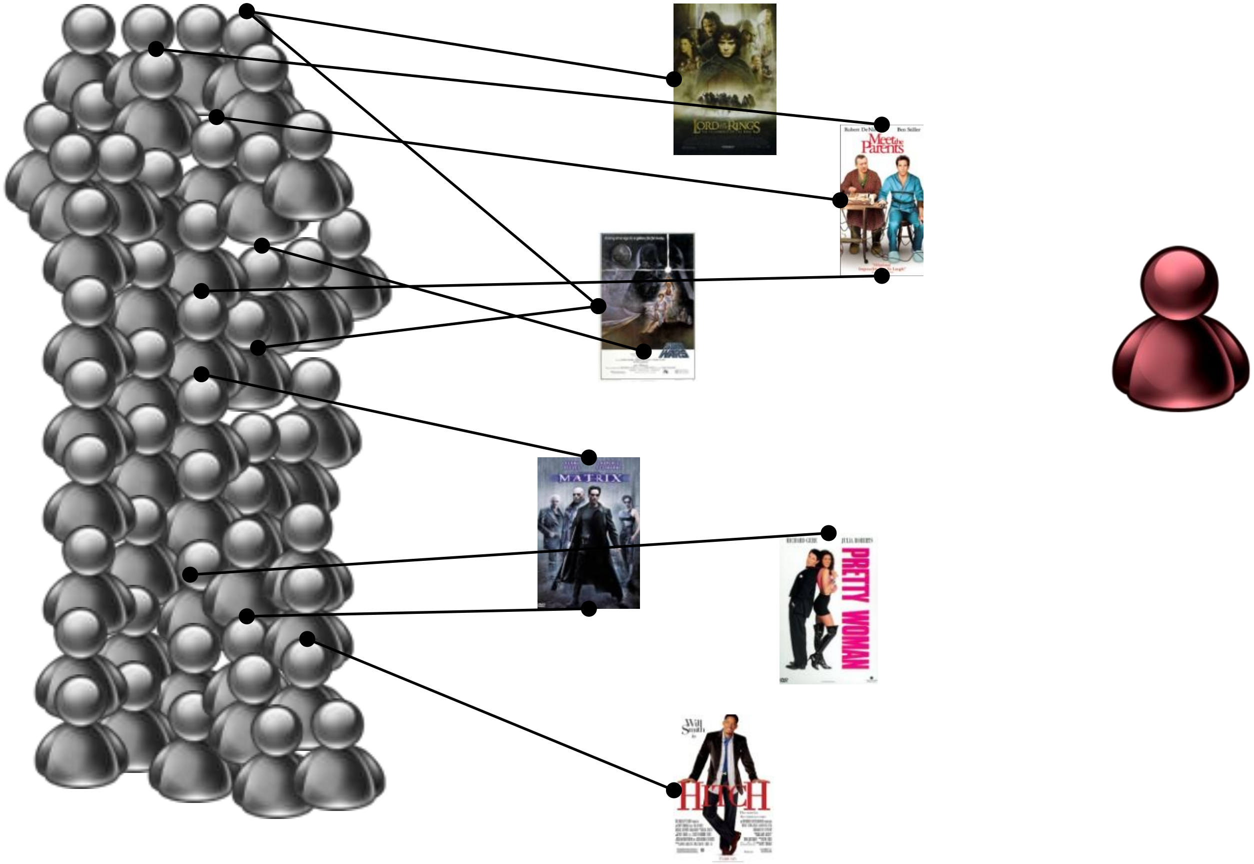
Context

- Recommender systems
  - Users interact (rate, purchase, click) with items

  - Which items will the user **like**?

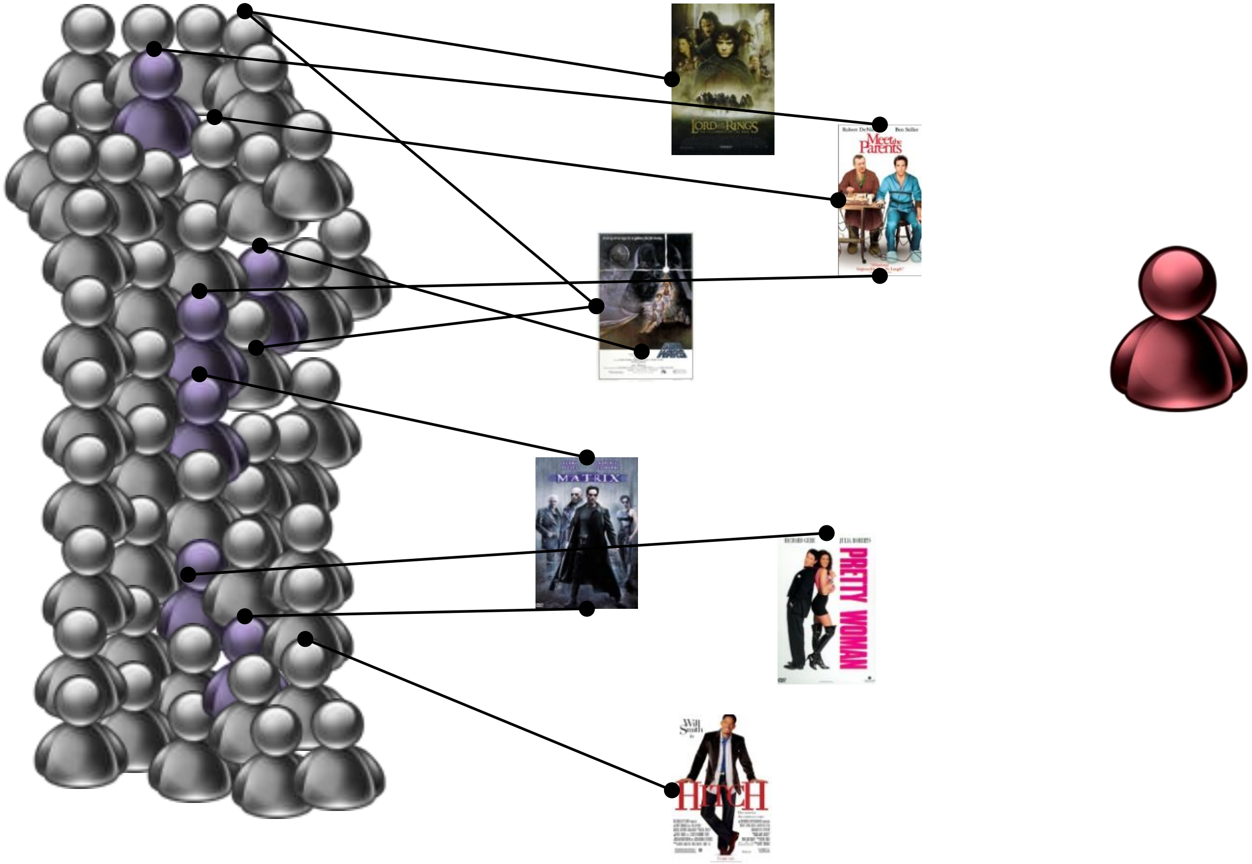
Context

- Nearest-neighbour recommendation methods
  - The item prediction is based on “similar” users

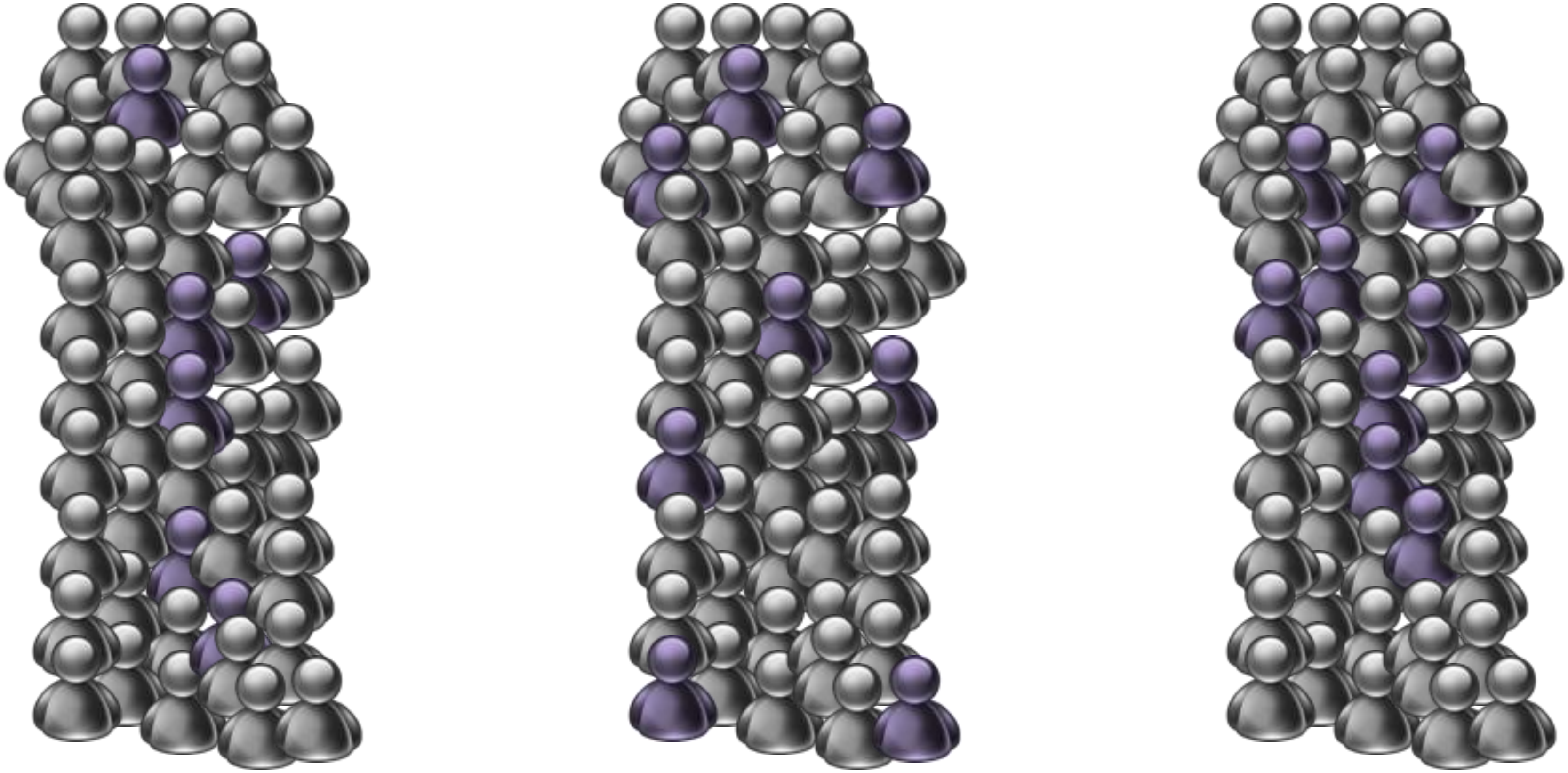


Context

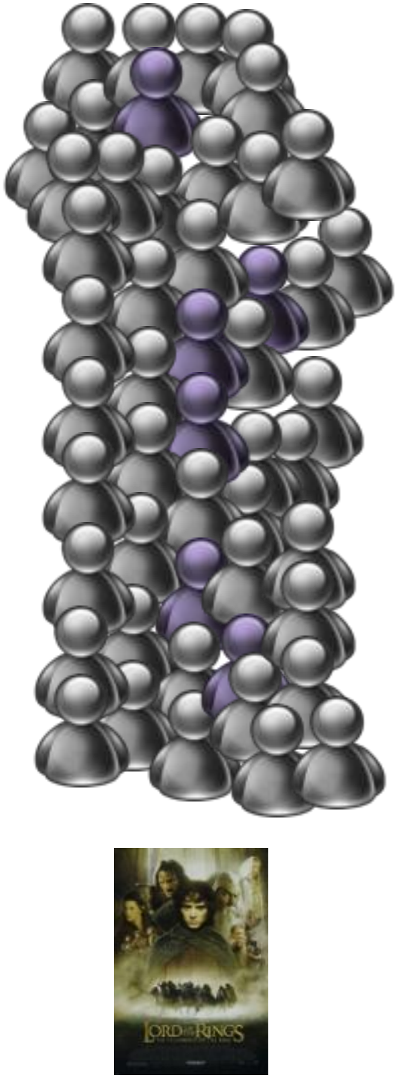
- Nearest-neighbour recommendation methods
  - The item prediction is based on “similar” users

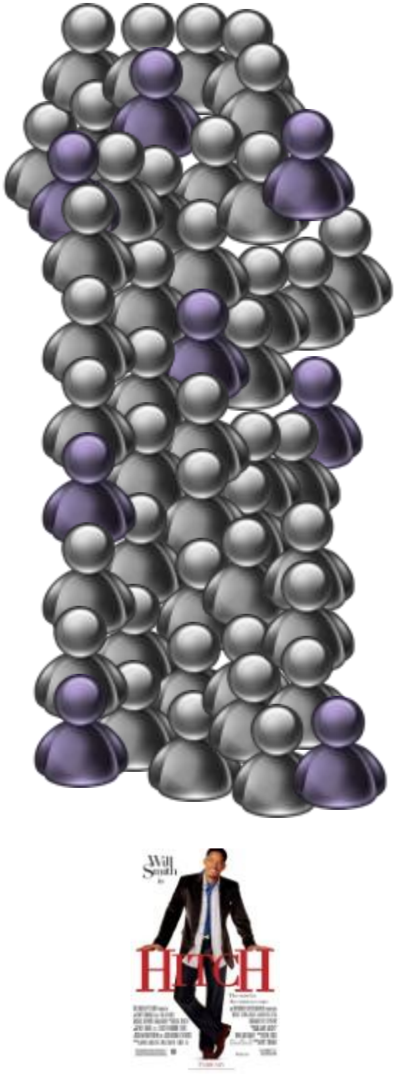


Different similarity metrics – different neighbours



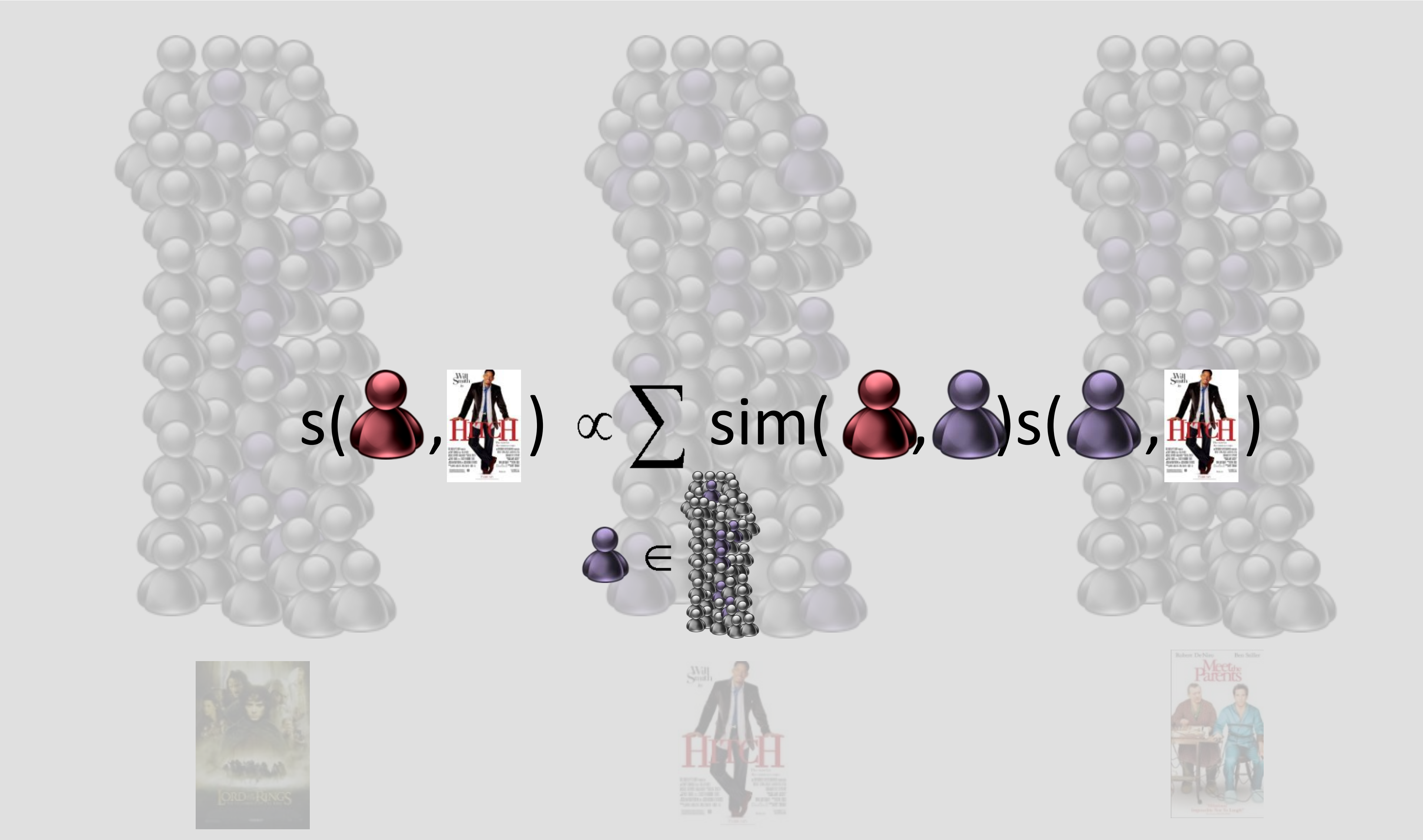
Different similarity metrics – different recommendations





Robert De Niro
Ben Stiller
Meet the Parents

Different similarity metrics – different recommendations



Research question

- How does the choice of a similarity metric determine the quality of the recommendations?

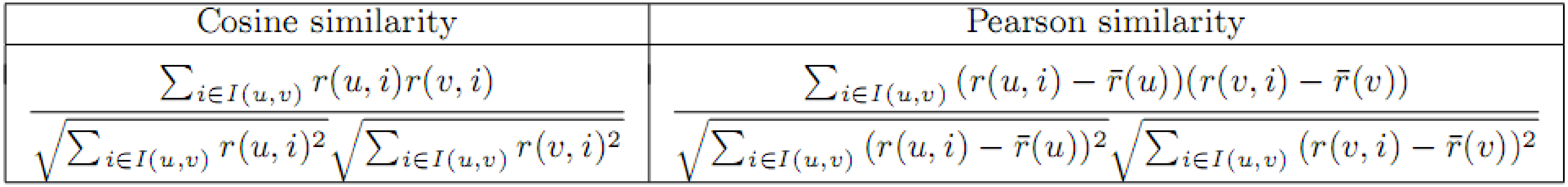
Problem: sparsity

- Too many items exist, not enough ratings will be available

- A user's neighbourhood is likely to introduce not-so-similar users

Different similarity metrics – which one is better?

- Consider Cosine vs Pearson similarity

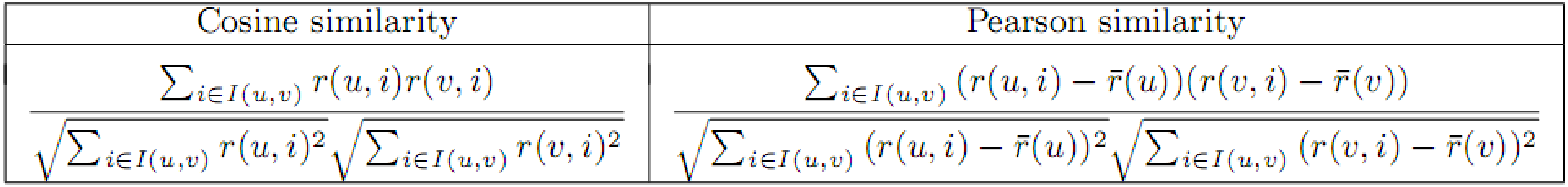


Cosine similarity	Pearson similarity
$\frac{\sum_{i\in I(u,v)} r(u,i)r(v,i)}{\sqrt{\sum_{i\in I(u,v)} r(u,i)^2}\sqrt{\sum_{i\in I(u,v)} r(v,i)^2}}$	$\frac{\sum_{i\in I(u,v)} (r(u,i)-\bar{r}(u))(r(v,i)-\bar{r}(v))}{\sqrt{\sum_{i\in I(u,v)} (r(u,i)-\bar{r}(u))^2}\sqrt{\sum_{i\in I(u,v)} (r(v,i)-\bar{r}(v))^2}}$

- Most existing studies report Pearson correlation to lead superior recommendation accuracy

Different similarity metrics – which one is better?

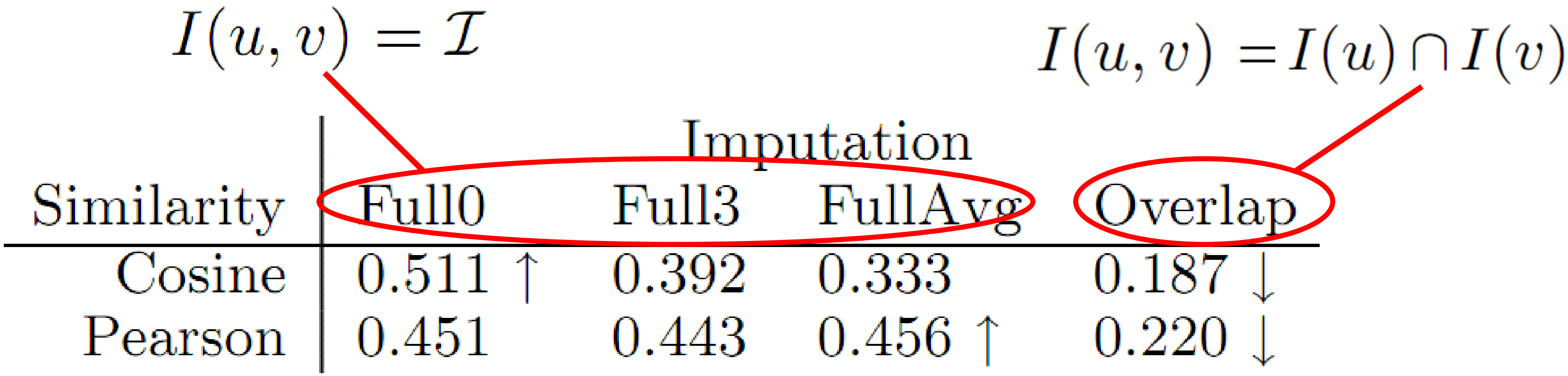
- Consider Cosine vs Pearson similarity



Cosine similarity	Pearson similarity
$\frac{\sum_{i \in I(u,v)} r(u,i) r(v,i)}{\sqrt{\sum_{i \in I(u,v)} r(u,i)^2} \sqrt{\sum_{i \in I(u,v)} r(v,i)^2}}$	$\frac{\sum_{i \in I(u,v)} (r(u,i) - \bar{r}(u))(r(v,i) - \bar{r}(v))}{\sqrt{\sum_{i \in I(u,v)} (r(u,i) - \bar{r}(u))^2} \sqrt{\sum_{i \in I(u,v)} (r(v,i) - \bar{r}(v))^2}}$

- Common variations to deal with sparsity
  - Thresholding: threshold to filter out similarities (no observed difference)
  - Item selection: use full profiles or only the overlap
  - Imputation: default value for unrated items

Different similarity metrics – which one is better?



$I(u,v) = \mathcal{I}$

$I(u,v) = I(u) \cap I(v)$

Similarity	Imputation			
	Full0	Full3	FullAvg	Overlap
Cosine	0.511 ↑	0.392	0.333	0.187 ↓
Pearson	0.451	0.443	0.456 ↑	0.220 ↓

- Which similarity metric is better?
  - Cosine is not superior for every variation
- Which variation is better?
  - They do not show consistent results
- Why some variations improve/decrease performance?

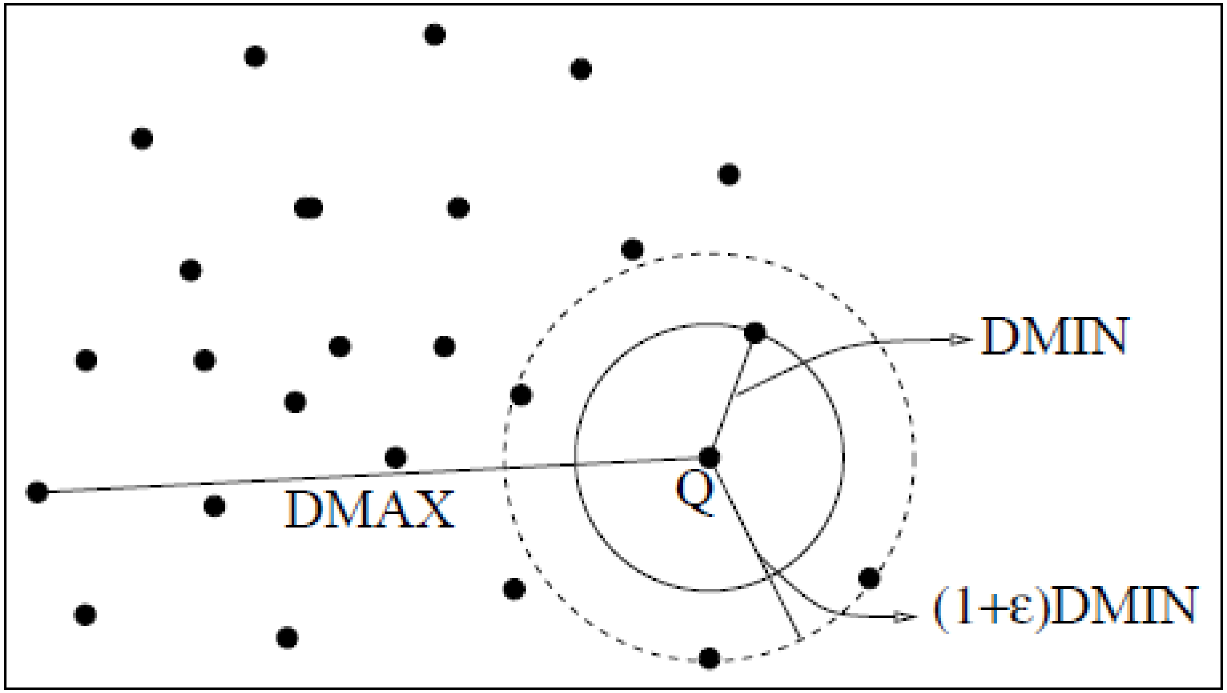
→ **Analysis of similarity features**

Analysis of similarity metrics

- Based on
  - Distance/Similarity distribution
  - Nearest-neighbour graph

Analysis of similarity metrics

- Distance distribution



- In high dimensions, nearest neighbour is ***unstable:***

  If the distance from query point to most data points is less than (1 + ε) times the distance from the query point to its nearest neighbour

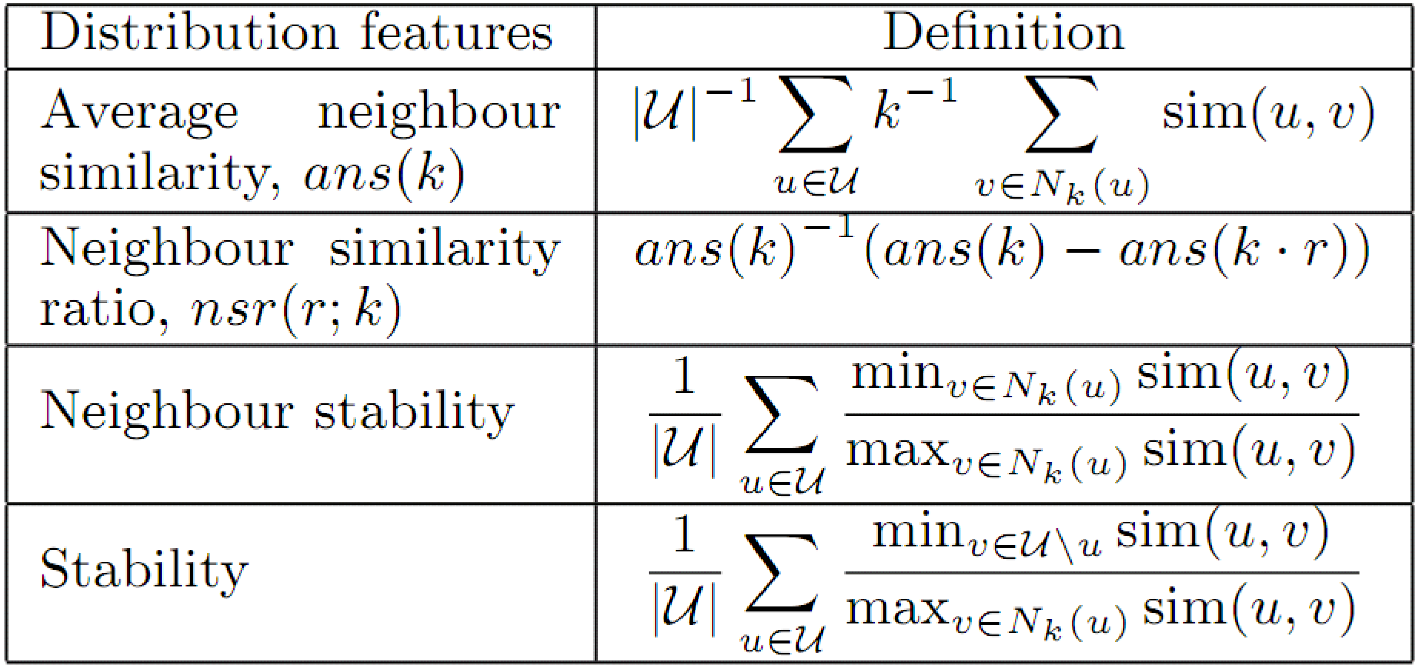
Beyer et al. *When is "nearest neighbour" meaningful?* ICDT 1999

Analysis of similarity metrics

- Distance distribution
  - Quality *q(n, f)*: fraction of users for which the similarity function has ranked at least *n* percentage of the whole community within a factor *f* of the nearest neighbour's similarity value

Analysis of similarity metrics

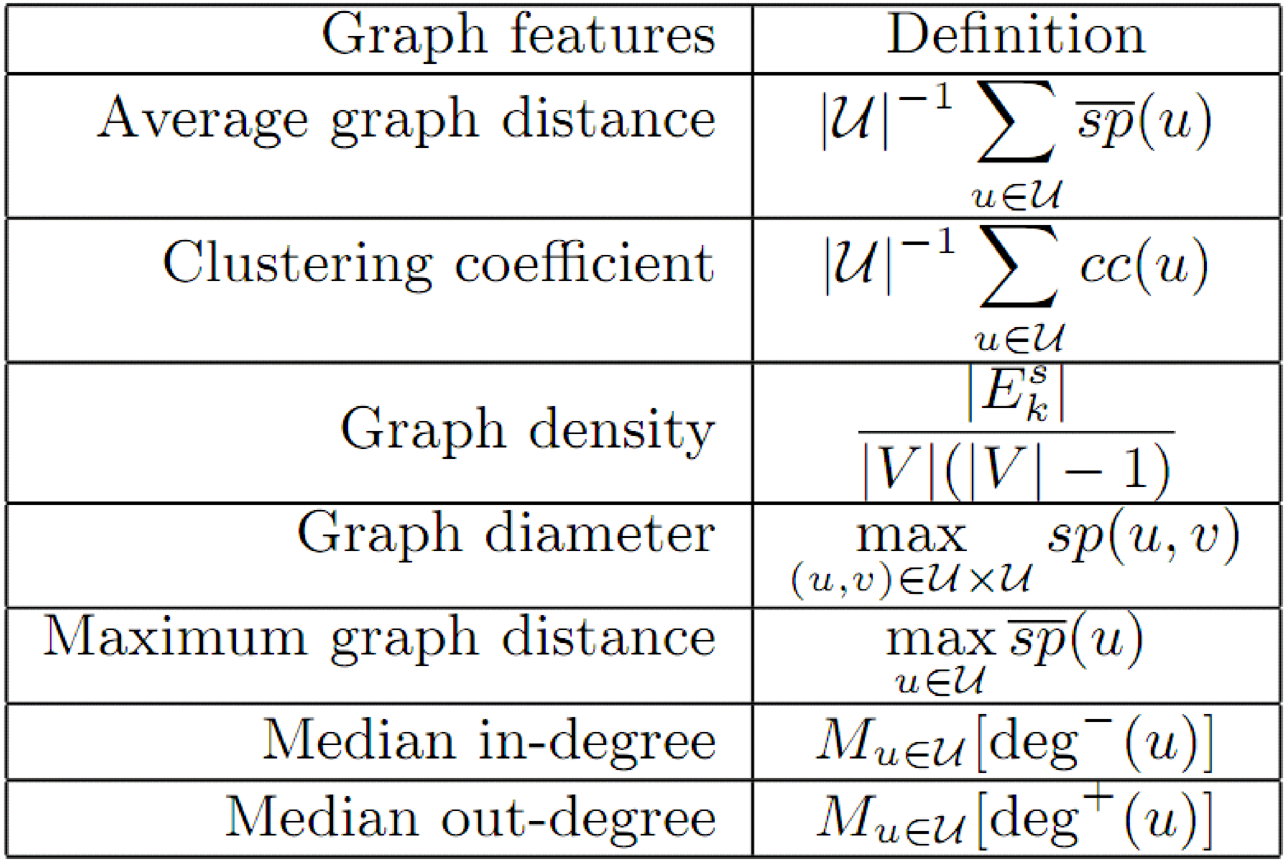
- Distance distribution
  - Quality *q(n, f)*: fraction of users for which the similarity function has ranked at least *n* percentage of the whole community within a factor *f* of the nearest neighbour’s similarity value
  - Other features:



Distribution features	Definition
Average neighbour similarity, $ans(k)$	$\lvert\mathcal{U}\rvert^{-1}\sum_{u\in\mathcal{U}} k^{-1}\sum_{v\in N_k(u)}\operatorname{sim}(u,v)$
Neighbour similarity ratio, $nsr(r;k)$	$ans(k)^{-1}(ans(k) - ans(k\cdot r))$
Neighbour stability	$\frac{1}{\lvert\mathcal{U}\rvert}\sum_{u\in\mathcal{U}}\frac{\min_{v\in N_k(u)}\operatorname{sim}(u,v)}{\max_{v\in N_k(u)}\operatorname{sim}(u,v)}$
Stability	$\frac{1}{\lvert\mathcal{U}\rvert}\sum_{u\in\mathcal{U}}\frac{\min_{v\in\mathcal{U}\setminus u}\operatorname{sim}(u,v)}{\max_{v\in N_k(u)}\operatorname{sim}(u,v)}$

Analysis of similarity metrics

- Nearest neighbour graph (NN^k)
  - Binary relation of whether a user belongs or not to a neighbourhood



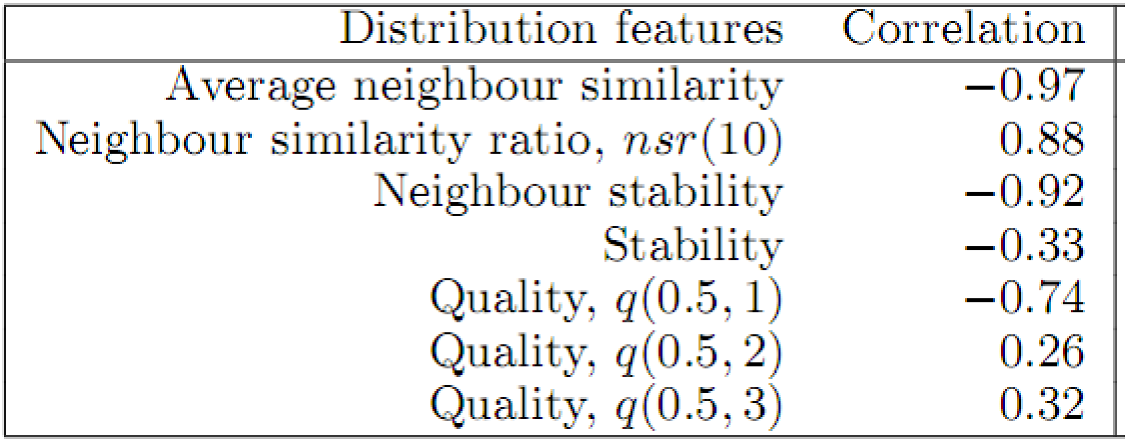
Graph features	Definition
Average graph distance	$\|\mathcal{U}\|^{-1} \sum_{u \in \mathcal{U}} \overline{sp}(u)$
Clustering coefficient	$\|\mathcal{U}\|^{-1} \sum_{u \in \mathcal{U}} cc(u)$
Graph density	$\frac{\|E_k^s\|}{\|V\|(\|V\| - 1)}$
Graph diameter	$\max_{(u,v) \in \mathcal{U} \times \mathcal{U}} sp(u, v)$
Maximum graph distance	$\max_{u \in \mathcal{U}} \overline{sp}(u)$
Median in-degree	$M_{u \in \mathcal{U}}[\deg^-(u)]$
Median out-degree	$M_{u \in \mathcal{U}}[\deg^+(u)]$

Experimental setup

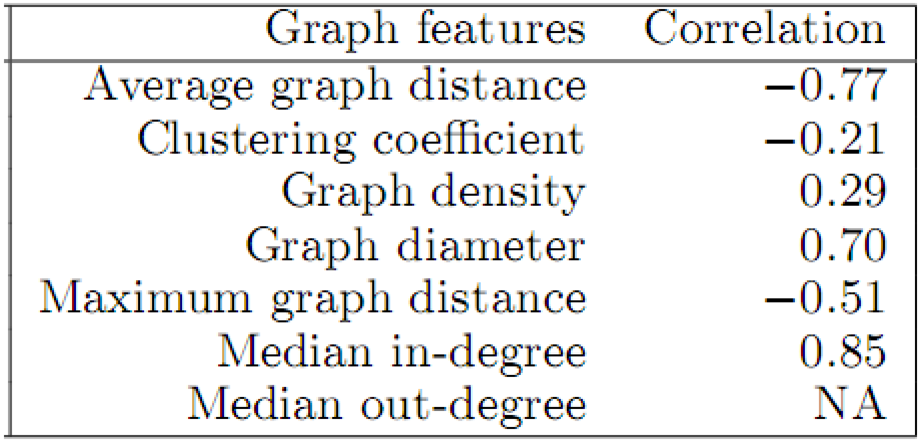
- Dataset
  - MovieLens 1M: 6K users, 4K items, 1M ratings
  - Random 5-fold training/test split
- JUNG library for graph related metrics
- Evaluation
  - Generate a ranking for each relevant item, containing 100 not relevant items
  - Metric: mean reciprocal rank (MRR)

Performance analysis

- Correlations between performance and features of each similarity (and its variations)



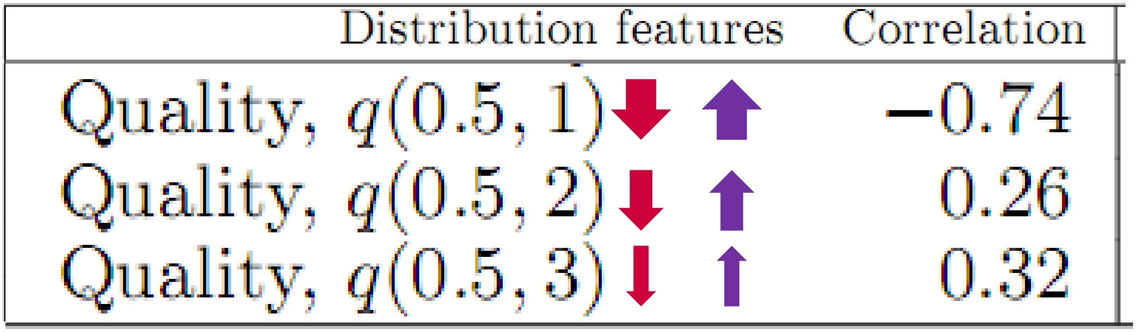
Distribution features	Correlation
Average neighbour similarity	−0.97
Neighbour similarity ratio, $nsr(10)$	0.88
Neighbour stability	−0.92
Stability	−0.33
Quality, $q(0.5, 1)$	−0.74
Quality, $q(0.5, 2)$	0.26
Quality, $q(0.5, 3)$	0.32



Graph features	Correlation
Average graph distance	−0.77
Clustering coefficient	−0.21
Graph density	0.29
Graph diameter	0.70
Maximum graph distance	−0.51
Median in-degree	0.85
Median out-degree	NA

Performance analysis – quality

- Correlations between performance and characteristics of each similarity (and its variations)

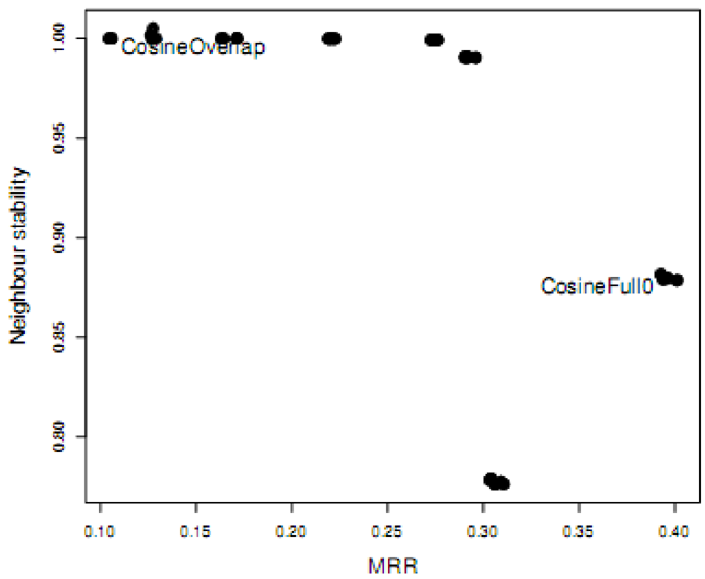


Distribution features	Correlation
Quality, $q(0.5, 1)$ ↓ ↑	-0.74
Quality, $q(0.5, 2)$ ↓ ↑	0.26
Quality, $q(0.5, 3)$ ↓ ↑	0.32

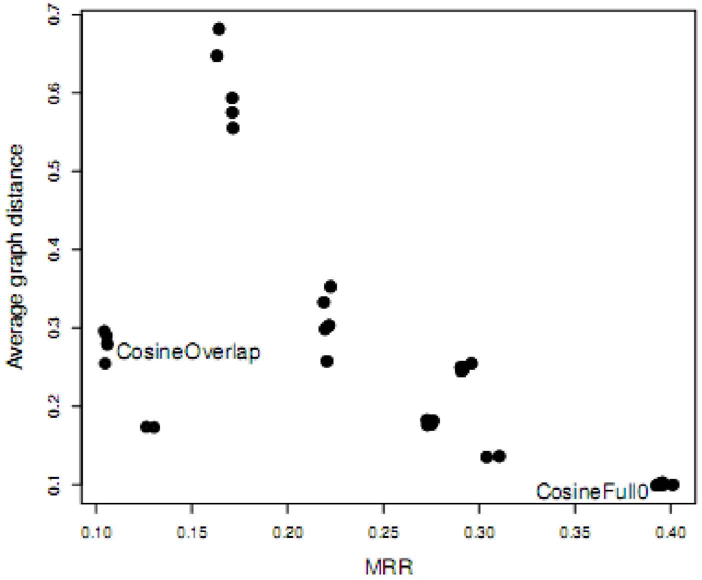
- For a user
  - If most of the user population is far away, low quality correlates with ***effectiveness*** (discriminative similarity)
  - If most of the user population is close, high quality correlates with ***ineffectiveness*** (not discriminative enough)

Quality *q(n, f)*: fraction of users for which the similarity function has ranked at least *n* percentage of the whole community within a factor *f* of the nearest neighbour's similarity value

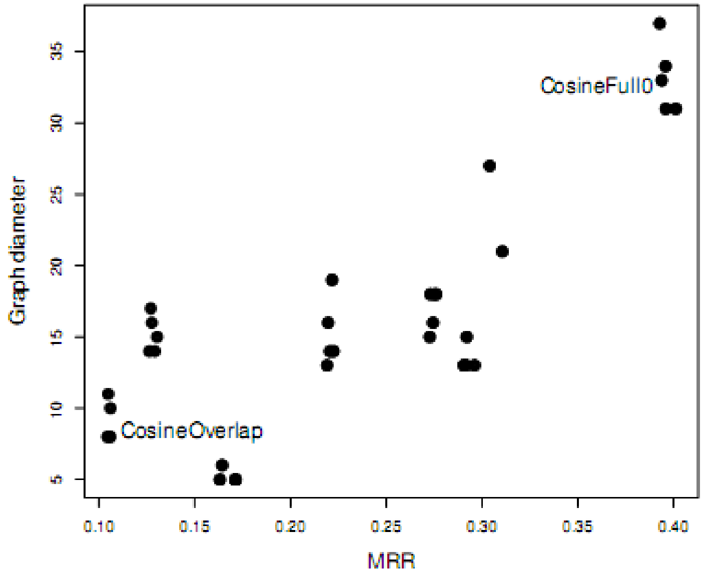
Performance analysis – examples



−0.92



−0.77



0.70

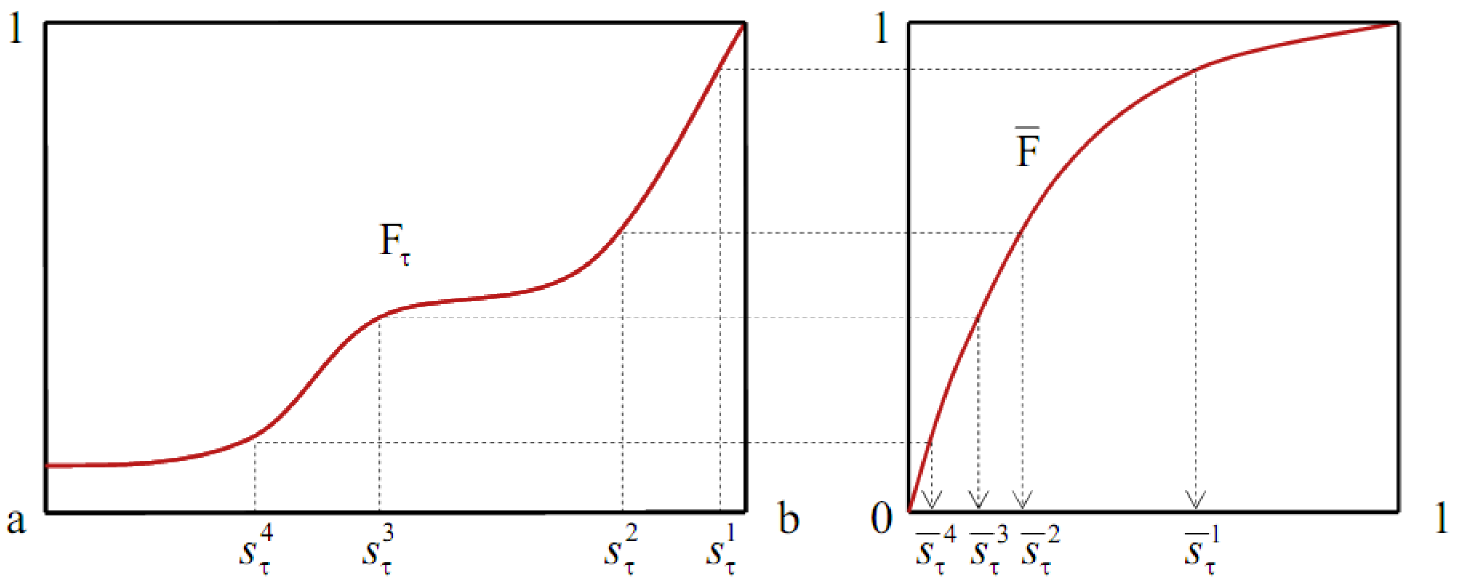
	Imputation			
Similarity	Full0	Full3	FullAvg	Overlap
Cosine	0.511 ↑	0.392	0.333	0.187 ↓
Pearson	0.451	0.443	0.456 ↑	0.220 ↓

Conclusions (so far)

- We have found similarity features correlated with their final performance
  - They are global properties, in contrast with query performance predictors
  - Compatible results with those in database: the stability of a metric is related with its ability to discriminate between good and bad neighbours

Application

- Transform “bad” similarity metrics into “better performing” ones
  - Adjusting their values according to the correlations found
- Transform their distributions
  - Using a distribution-based normalisation [Fernández, Vallet, Castells, ECIR 06]
  - Take as ideal distribution ($\bar{F}$) the best performing similarity (Cosine Full0)



$$\bar{s}_\tau^i = \bar{F}^{-1} \circ F_\tau\left(s_\tau^i\right)$$

$$s_\tau^i = s_\tau\left(x_i\right)$$

$$\bar{s}_\tau^i = \bar{s}_\tau\left(x_i\right)$$

Application

- Transform “bad” similarity metrics into “better performing” ones
  - Adjusting their values according to the correlations found
- Transform their distributions
  - Using a distribution-based normalisation [Fernández, Vallet, Castells, ECIR 06]
  - Take as ideal distribution ($\overline{\mathrm{F}}$) the best performing similarity (Cosine Full0)
- Results

	Top 50	
	Orig	Norm
Cosine Full0	0.511	-
Cosine Full3	0.392	0.388
Cosine FullAvg	0.333	0.328
Cosine Overlap	0.187	0.181
Pearson Full0	0.451	0.445
Pearson Full3	0.443	0.443
Pearson FullAvg	0.456	0.454
Pearson Overlap	0.220	0.217

The rest of the characteristics are not (necessarily) inherited

Conclusions

- We have found similarity features correlated with their final performance
  - They are global properties, in contrast with query performance predictors
  - Compatible results with those in database: the stability of a metric is related with its ability to discriminate between good and bad neighbours

- Not conclusive results when transforming bad-performing similarities based on distribution normalisations
  - We want to explore (and adapt to) other features, e.g., graph distance
  - We aim to develop other applications based on these results, e.g., hybrid recommendation

Thank you

Understanding Similarity Metrics in Neighbour-based Recommender Systems

Alejandro Bellogín, Arjen de Vries

Information Access

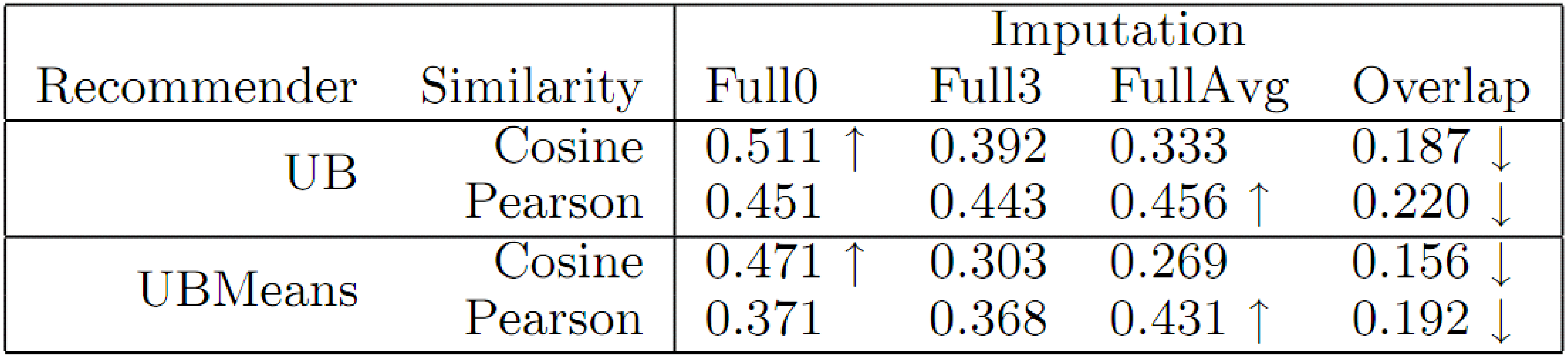
CWI

ICTIR, October 2013

Different similarity metrics – all the results

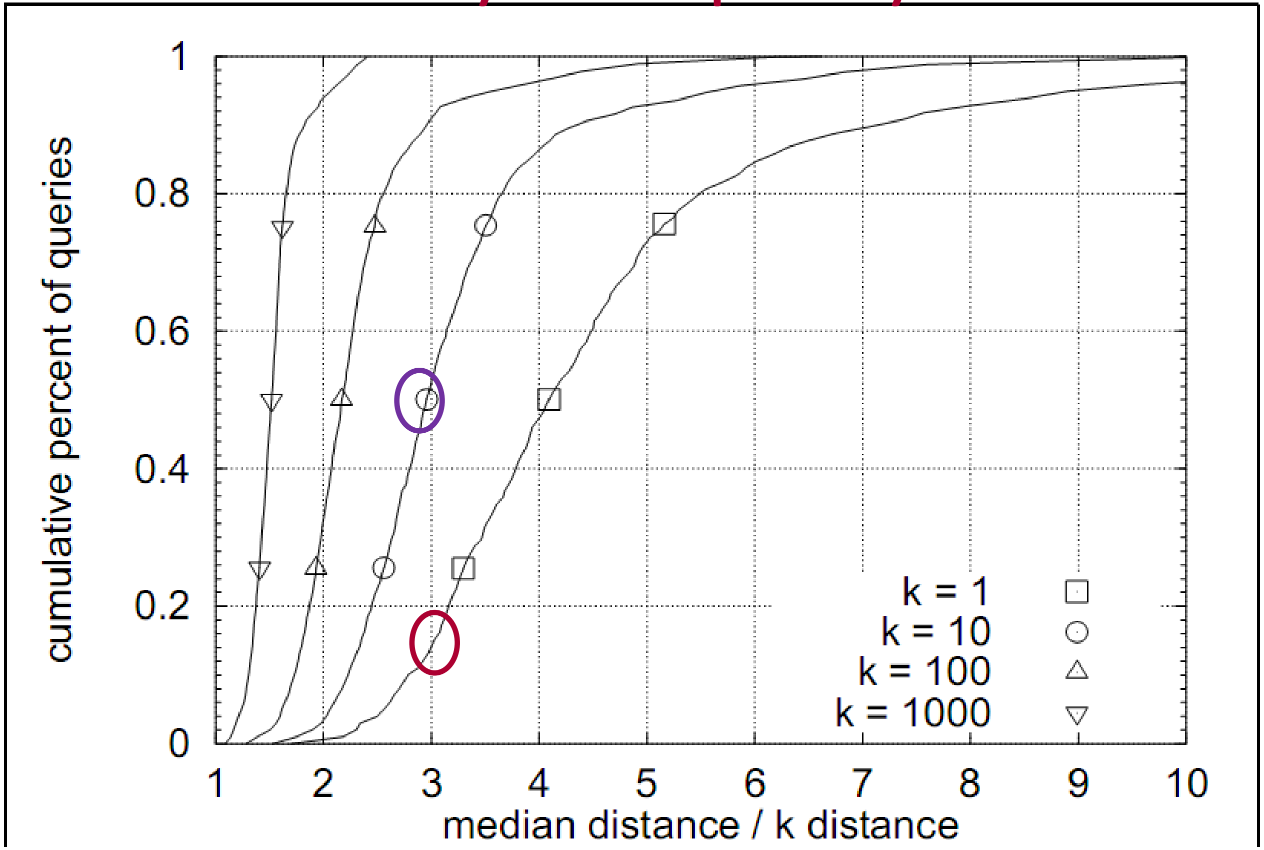
- Performance results for variations of two metrics
  - Cosine
  - Pearson

- Variations
  - Thresholding: threshold to filter out similarities (no observed difference)
  - Imputation: default value for unrated items



		Imputation			
Recommender	Similarity	Full0	Full3	FullAvg	Overlap
UB	Cosine	0.511 ↑	0.392	0.333	0.187 ↓
	Pearson	0.451	0.443	0.456 ↑	0.220 ↓
UBMeans	Cosine	0.471 ↑	0.303	0.269	0.156 ↓
	Pearson	0.371	0.368	0.431 ↑	0.192 ↓

Beyer’s “quality”



To determine the quality of answers for NN queries, we examined the percentage of queries in which at least half the data points were within some factor of the nearest neighbor. Examine the graph at *median distance/k distance* = 3. The graph says that for $k = 1$ (normal NN problem), 15% of the queries had at least half the data within a factor of 3 of the distance to the NN. For $k = 10$, 50% of the queries had at least half the data within a factor of 3 of the distance to the 10th nearest neighbor. It is easy to see that the effect of changing k on the quality of the answer is most significant for small values of k.