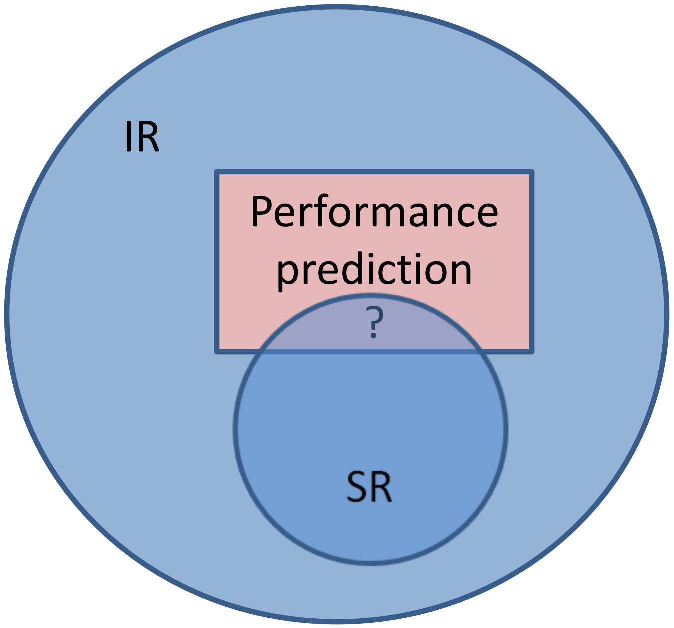
Performance prediction in recommender systems: application to the dynamic optimisation of aggregative methods

Alejandro Bellogín Kouki

Advisor: Pablo Castells Azpilicueta

Departamento de Ingeniería Informática
Escuela Politécnica Superior
Universidad Autónoma de Madrid

Introduction



IR : Information Retrieval

RS : Recommender Systems

Outline

- Motivation and goals
- Related work and context
- Formal framework
  - Dynamic aggregation in recommender systems
  - Performance prediction for recommender systems' components
- Neighbour weighting in collaborative filtering
  - Description
  - Experiments
- Weighting in hybrid recommendation
  - Description
  - Experiments
- Conclusion

Motivation

- Proliferation and variety of input information in IR systems

- Performance prediction in IR: adjust retrieval strategies according to the value of the prediction function

  - In classic retrieval: query effectiveness

  - Applications: query expansion, rank fusion, distributed IR, etc.

Goals

- State of the art study and analysis on performance prediction in Information Retrieval
- Study the potential of performance prediction in Recommender Systems
- Definition of performance predictors in Recommender Systems
- Effective application of performance predictors to combined Recommender System methods

Performance prediction in IR: query expansion

- Deciding if a query has to be expanded or not, i.e., given a predictor γ and a query q:

  if $\gamma(q)$ is greater than a particular value (threshold) then the query will perform well, and it should not be expanded,

  otherwise, the query is expanded.
- Problems:
  - Predictor definition (the higher the value, the better the query performance)
  - Threshold value definition (optimum value can be found)
- Calculate the ambiguity, vagueness, or specificity of the query
  - q_A: “race”
  - q_B: “race car”, “race horse”

$$\gamma(q_B) > \gamma(q_A)$$

- SoA: Cronen-Townsend et al. 2002, He & Ounis 2004, Diaz & Jones 2004, Mothe & Tanguy 2005, Jensen et al. 2005, Amati et al. 2004, Zhou & Croft 2006, Zhou & Croft 2007, Carmel et al. 2006

Query performance predictors in IR

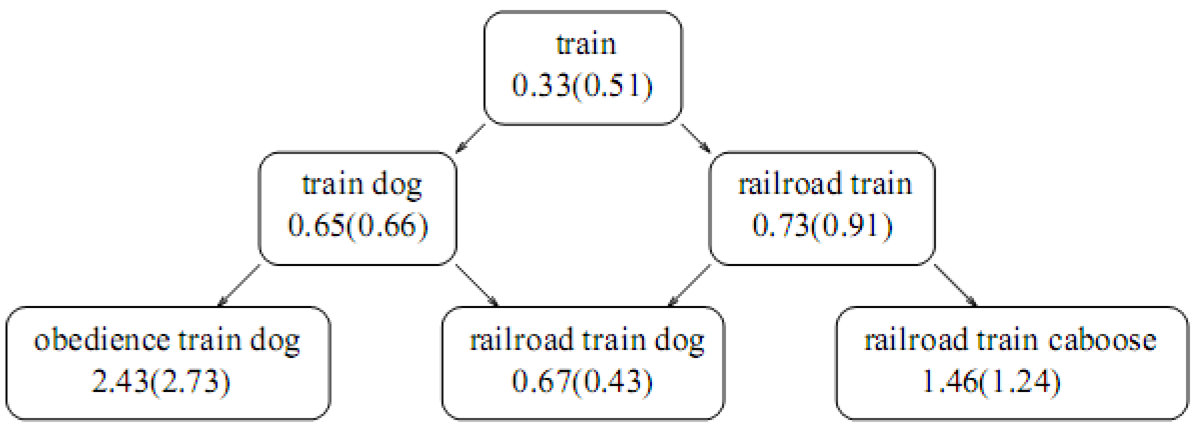
- Clarity: distance (relative entropy) between query and collection language models

$$\text{clarity}(q) = \sum_{w \in V} P(w|q) \log_2 \frac{P(w|q)}{P_{coll}(w)}$$

$$P(w|q) = \sum_{d \in R} P(w|d) P(d|q), \quad P(q|d) = \prod_{w_q \in q} P(w_q|d)$$

$$P(w|d) = \lambda P_{ml}(w|d) + (1-\lambda) P_{coll}(w)$$

- Example:



Performance prediction in IR: rank fusion

- Application to rank fusion
  - Return a single aggregated retrieval result set from different input rankings (p.e., metasearch, distributed search, etc.) (Fox & Shaw 1993)
- In this area, performance prediction is used for weighting each source to be combined (according to their estimated performance)
- SoA: Yom-Tov et al. 2005, Aslam & Pavlu 2007, Castells et al. 2005, Diaz 2007
- Example:
  - "flu" ➔ 0.8 * PubMed + 0.2 * MathSciNet
  - "integral" ➔ 0.2 * PubMed + 0.8 * MathSciNet

Context: recommender systems

- In this work: application to recommender systems

objects

users

	i_1		i_k		i_m
u_1	r_{11}		r_{1k}		r_{1m}
u_j	r_{j1}		?		r_{jm}
u_n	r_{n1}		r_{nk}		r_{nm}

how is predicted the value of r_{jk}?

Context: recommender systems

- Collaborative filtering (CF) based on users

items

users

	i_1		i_k		i_m
u_1	r_{11}		r_{1k}		r_{1m}
u_j	r_{j1}		?		r_{jm}
u_n	r_{n1}		r_{nk}		r_{nm}

Context: recommender systems

- Collaborative filtering (CF) based on items

items

users

	$\mathbf{i_1}$		$\mathbf{i_k}$		$\mathbf{i_m}$
$\mathbf{u_1}$	r_{11}		r_{1k}		r_{1m}
$\mathbf{u_j}$	r_{j1}		?		r_{jm}
$\mathbf{u_n}$	r_{n1}		r_{nk}		r_{nm}

Context: recommender systems

- Content-based recommendation (CB)

items

users

	i_1		i_k		i_m
u_1	r_{11}		r_{1k}		r_{1m}
u_j	r_{j1}		?		r_{jm}
u_n	r_{n1}		r_{nk}		r_{nm}

Context: recommender systems

- CF and CB both have disadvantages:

Problem	**Description**	**CF**	**CB**
Grey sheep	A user with different taste with respect to the rest of the community	X	
Sparsity	The number of available ratings is small	X	
New item	Recommender items must be rated by a high number of users	X	
New user	Each user has to rate enough items in order to infer her preferences	X	X
Content analysis	Recommended items must have available data about their properties		X
Specialisation	Recommended items are very similar		X

- An alternative to these problems are hybrid strategies:
  - Cascade
  - Weighting (linear combination or voting schemes)

Aggregation operations in RS

- Producing recommendations in CF

$$g(u_m, i_n) = \frac{\sum_{u_j \in N[u_m]} \operatorname{sim}(u_m, u_j) \times r_{j,n}}{\sum_{u_j \in N[u_m]} \left|\operatorname{sim}(u_m, u_j)\right|}$$

- Hybrid recommendation

$$g(u_m, i_k) = \lambda \times g_{CB}(u_m, i_k) + (1 - \lambda) \times g_{CF}(u_m, i_k)$$

- Aggregation operation in personalized IR

$$s(d) = \lambda \times s_q(d) + (1 - \lambda) \times s_P(d)$$

- Research question:

  Is it feasible to apply prediction performance to these problems?

How?

Formal framework

- Utility Theory: we formulate the IR problem as

$$\text{find } g : \mathcal{D} \times \Omega \rightarrow \mathbb{R}, \text{ such as } d_1 \leq_q d_2 \Leftrightarrow g(d_1, q) \leq g(d_2, q)$$

- IR systems are built as a high-level composition, where each subcomponent implements different criteria or strategies:

$$\vec{g} = g(d, q) = \varphi\,(g_1(d, q), \ldots, g_n(d, q))$$

- Linear combination is a fairly general aggregation function in RI:

$$\varphi\,(s_1, \ldots, s_n) = \alpha_1\, s_1 + \cdots + \alpha_n\, s_n = \vec{\alpha} \bullet \vec{s}$$

- Examples:
  - Rating prediction in CF
  - Hybrid recommendation
  - Personalised search (linear combination between ad hoc retrieval and personalised scores) : $\lambda \times \text{score}_q\,(d) + (1 - \lambda) \times \text{score}_p\,(d)$
  - Rank fusion: $\text{score}(d) = \Sigma_\tau\, \text{score}_\tau\,(d)$ (CombSUM)

Formal framework

- This combination can be improved by using the best possible weighting parameters, i.e.:

$$\vec{\alpha}^* = \arg\max_{\vec{\alpha} \in \mathbb{R}^n} \left(\rho\left(\vec{g} \right) \right)$$

  - ρ represents a quality measure of the decision encoded by $\vec{g}$
- Coefficient α_j determines the dominance that each g_j shall have in the aggregated decision $\vec{g}$
- In this way, we can gain performance of g favouring the influence of the g_j that supply a better quality output in each situation

- We need to "predict" the performance of each component:
  - Function (or predictor) $\gamma_j\,(d, q, \Omega)$
- Thus: $\alpha_j = \psi(\gamma_j\,(d, q, \Omega))$
  - Simplification: $\psi(x) = x$

Hypotheses

1. A linear combination is a good and general function to build composite IR systems
2. A suitable dynamic assignment of combination weights enables an improved performance in a combined retrieval system
3. Performance predictors from the IR field can be adapted to RS and result in effective predictors
4. The effectiveness of performance predictors can be assessed by their correlation to suitable performance metrics
5. The performance of an IR system, or component, is monotonically decreasing with the amount of uncertainty involved in the retrieval problem at hand

Dynamic fusion in Recommender Systems

- In this case: $\mathcal{D} = \mathcal{I}$ y $\Omega = (\mathcal{U}, r)$, i.e., the retrieval space is the set of all possible items that can be recommended, the input space is the set of all users and $r : \mathcal{S} \to \mathcal{R}$ with $\mathcal{S} \subset \mathcal{U} \times \mathcal{I}$, provides a set of user ratings for items (which indicates how a particular user liked a particular item)

- Sometimes the recommender wants to find the best item (top 1):

$$g : \mathcal{U} \times \mathcal{I} \to \mathcal{R}$$

$$\forall\, \mathrm{u} \in \mathcal{U},\ \mathrm{i}_{\mathrm{u}}^{*} = \arg\max_{\mathrm{i} \in \mathcal{I}} g\,(\mathrm{u}, \mathrm{i})$$

- The utility of an item is equated to a rating (actual or predicted by the system)

Performance predictors for Recommender Systems

- Based on clarity
  - User- vs. item-based
  - Using users (UUC, UIC) or items (IUC, IIC)

- Based on Information Theory
  - User (UIG) and item (IIG) information gain

- Heuristic
  - Count frequencies (inspired by IR's IDF):
    - User (IUF) and item (IIF) inverse frequency
    - User's (IURF) and item's (IIRF) rating inverse frequency
    - User's (UF) and item's (IF) features frequency

Neighbour weighting in Collaborative Filtering

- Formalisation:

$$g\left(u_m, i_k\right) = \varphi_{N[u_m]}\left(g_1\left(u_m, i_k\right), \cdots, g_n\left(u_m, i_k\right)\right)$$

$$g_j\left(u_m, i_k\right) = \text{sim}\left(u_m, v_j\right) \times r\left(u_j, i_k\right)$$

$$\alpha_j = \frac{\gamma_j}{\sum_{v \in N[u_m]} \left|\text{sim}\left(u_m, v\right)\right|}$$

$$g\left(u_m, i_n\right) = \frac{\sum_{u_j \in N[u_m]} \text{sim}\left(u_m, u_j\right) \times r_{j,n}}{\sum_{u_j \in N[u_m]} \left|\text{sim}\left(u_m, u_j\right)\right|}$$

Neighbour weighting in Collaborative Filtering

- Predictors used:
  - *Item-based user clarity* (IUC):

$$\gamma_j = \gamma_j(v_j) = \mathrm{IUC}(u) = \sum_{i \in I} p(i|u) \log_2 \frac{p(i|u)}{p_c(i)}$$

$$p(i|u) = \lambda \frac{rat(u,i)}{5} + (1-\lambda)\, p_c(i)$$

$$p_c(i) = \frac{1}{|I|}$$

  - *User-based user clarity* (UUC):

$$\gamma_j = \gamma_j(v_j) = \mathrm{UUC}(u) = \sum_{v \in U} p(v|u) \log_2 \frac{p(v|u)}{p_c(v)}$$

$$p(v|u) = \sum_{i : rat(u,i) \neq 0} p(v|i)\, p(i|u)$$

$$p(v|i) = \lambda \frac{rat(v,i)}{5} + (1-\lambda)\, p_c(v)$$

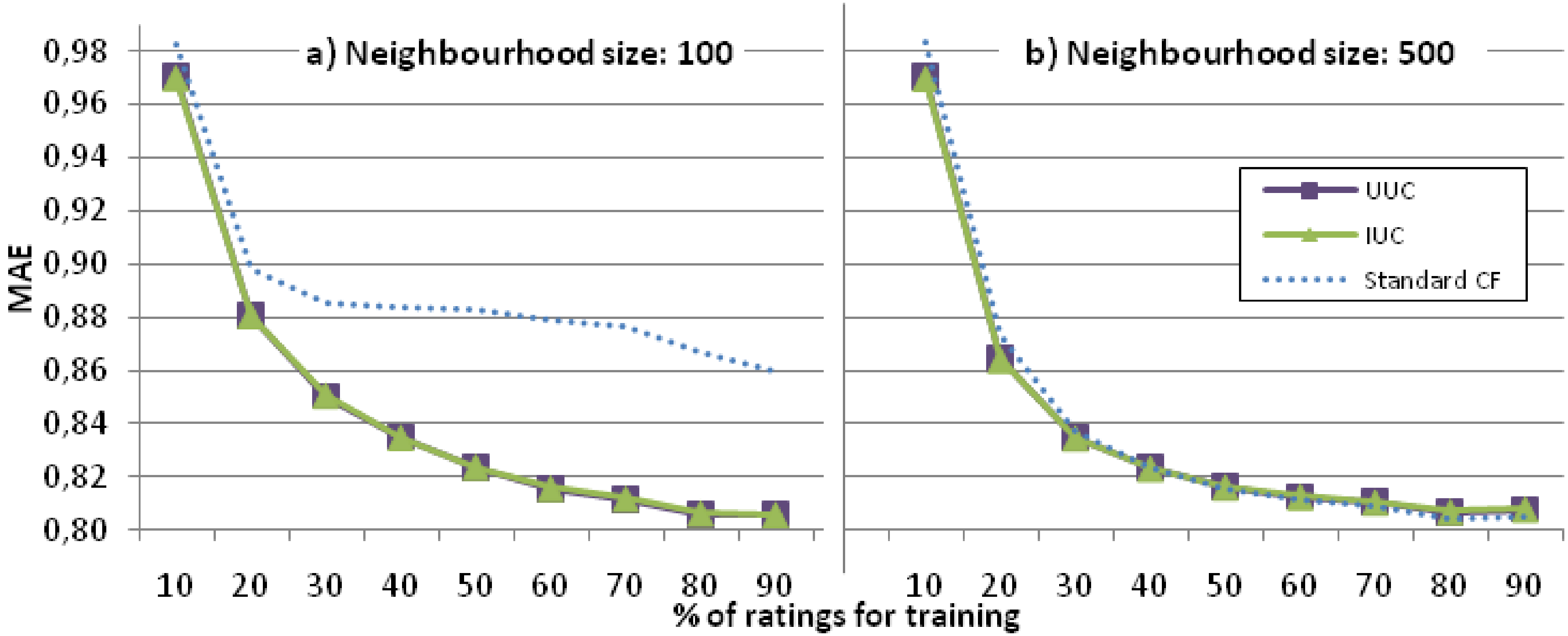
$$p_c(v) = \frac{1}{|U|}$$

Neighbour weighting in Collaborative Filtering

- Results:
  - Evaluation metric: MAE (mean average error)

$$\mathrm{MAE} = \frac{1}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N}\left| r_{m,n} - p_{m,n} \right|$$

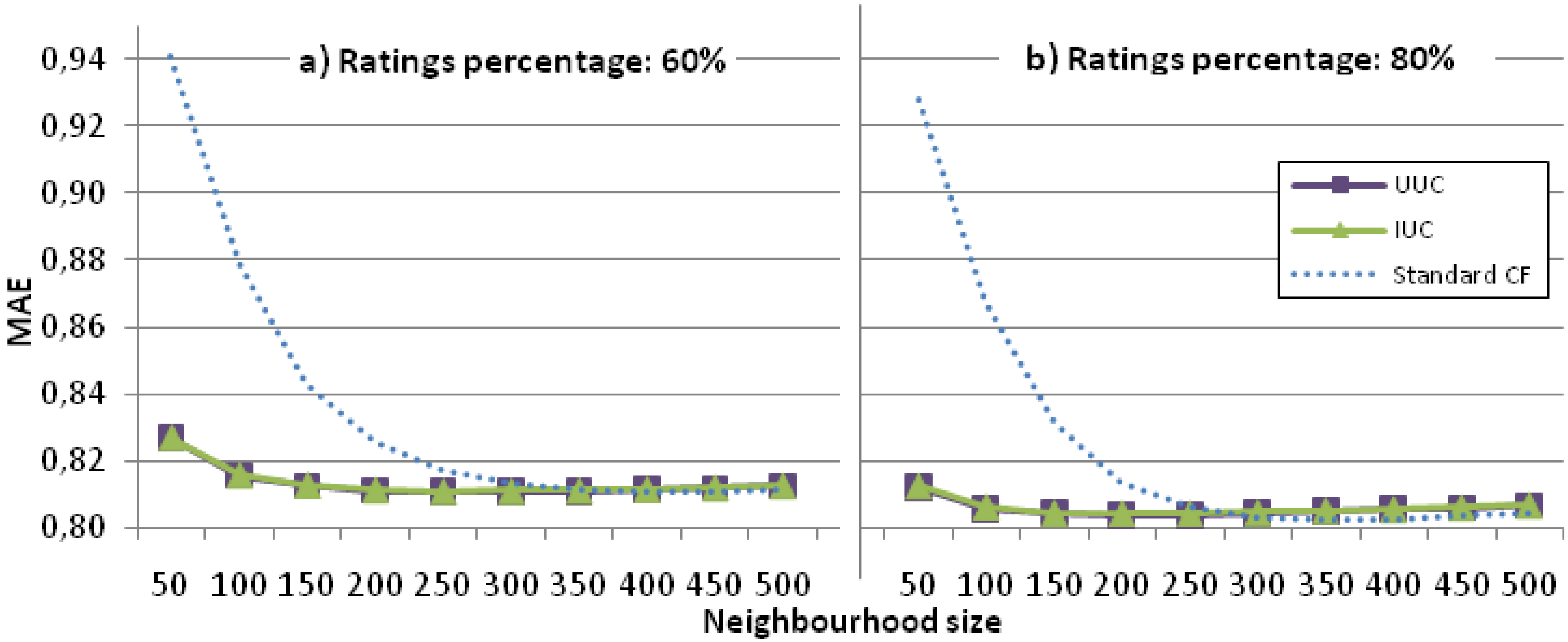
  - Performance comparison for different rating density:



Neighbour weighting in Collaborative Filtering

- Results (cont.):

  - Performance comparison for different neighbourhood sizes (size of N[u])



Neighbour weighting in Collaborative Filtering

- Results (cont.):
  - Correlation analysis with respect to a new metric: *Neighbour Goodness*

$$\text{neighbour\_goodness}(u) = \text{MAE}\left(\mathcal{U} - \{u\}, \mathcal{R} - \mathcal{R}(u)\right) - \text{MAE}\left(\mathcal{U} - \{u\}, \mathcal{R}\right)$$

  - Pearson's correlation statistical significant at a level of 5%

Performance predictor	**% of ratings**								
	10	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**
UUC	-0.23	0.21	0.26	0.22	0.21	0.20	0.19	0.18	0.15
IUC	-0.24	0.17	0.19	0.15	0.14	0.13	0.13	0.13	0.09

Weighted hybrid recommendation

- Formalisation:

$$g(u_m, i_k) = \alpha_{CB} \times g_{CB}(u_m, i_k) + \alpha_{CF} \times g_{CF}(u_m, i_k)$$

$$\alpha_{CB} = \gamma_{CB}$$

$$\alpha_{CF} = \gamma_{CF}$$

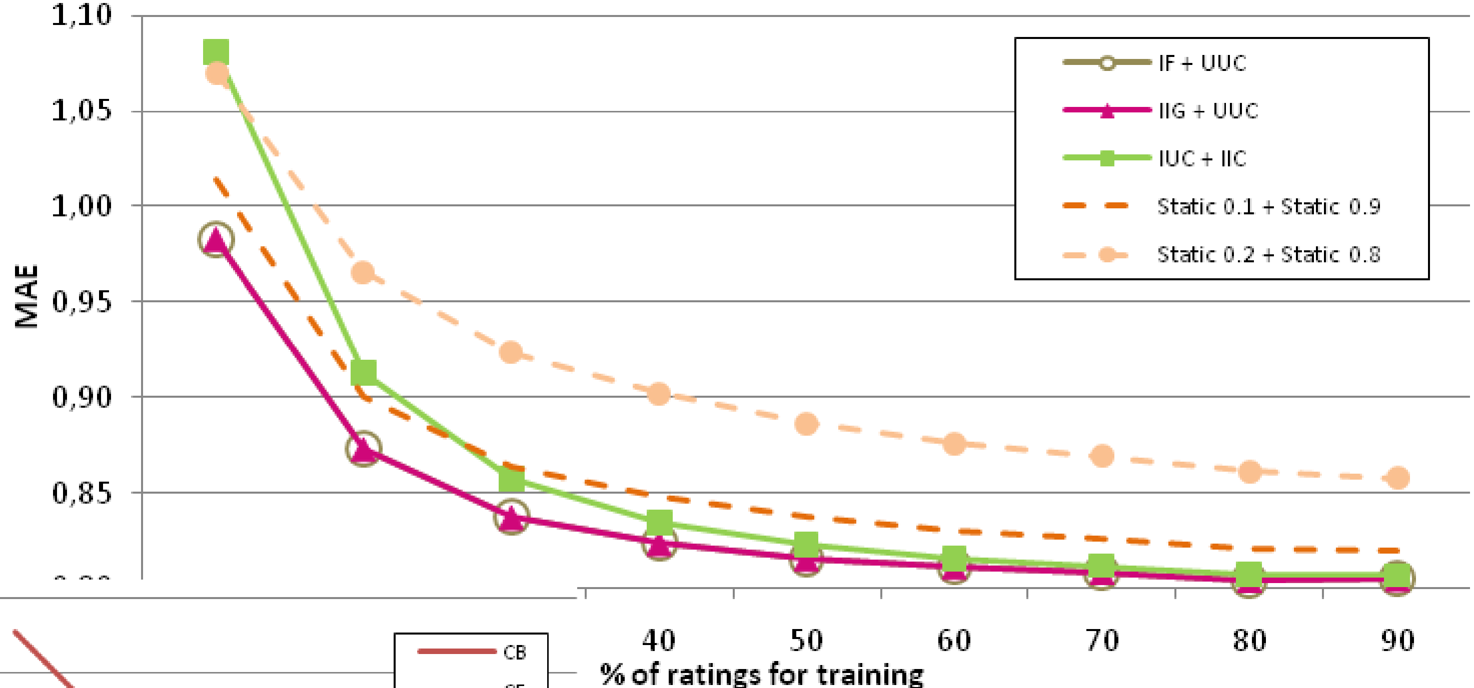
$$g(u_m, i_k) = \lambda \times g_{CB}(u_m, i_k) + (1 - \lambda) \times g_{CF}(u_m, i_k)$$

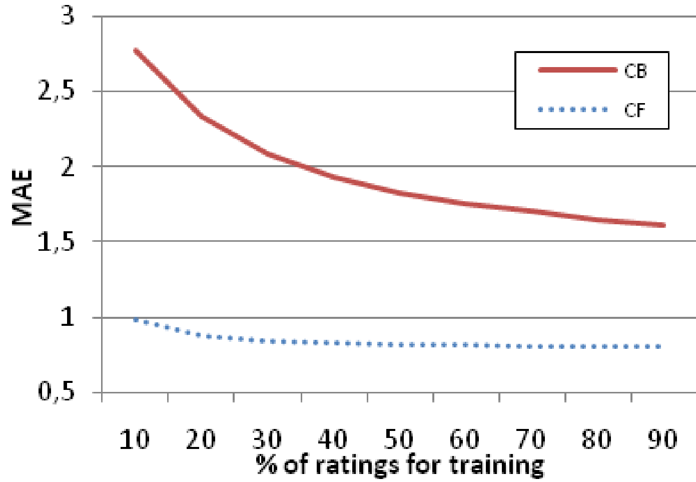
Weighted hybrid recommendation

- Predictors used:
  - Content-based component (γ_{CB})
    - *Item features* (IF): $$\gamma_{CB}=\gamma_{CB}(i_k)=\mathrm{IF}(i_k)=\sum_{t\in\mathfrak{T}}\text{TF-IDF}(i,t)$$
    - *Item information gain* (IIG): $$\gamma_{CB}=\gamma_{CB}(i_k)=\mathrm{IIG}(i_k)=\frac{p_1^i-p_0^i}{p_1^i}$$
    - *Item-based user clarity* (IUC): $$\gamma_{CB}=\gamma_{CB}(v_j)=\mathrm{IUC}(v_j)$$
  - Collaborative component (γ_{CF})
    - *Item-based item clarity* (IIC): $$\gamma_{CF}=\gamma_{CF}(i_k)=\mathrm{IIC}(i_k)=\sum_{u\in U}p(u\,|\,i)\log_2\frac{p(u\,|\,i)}{p_c(u)}$$
    - *User-based user clarity* (UUC): $$\gamma_{CF}=\gamma_{CF}(v_j)=\mathrm{UUC}(v_j)$$
- We also consider the static baseline as a predictor.
  - Static baseline, with parameter λ : $$\gamma_{CB}=\gamma_{CF}=\gamma^{\lambda}=\lambda$$

Weighted hybrid recommendation

- Results





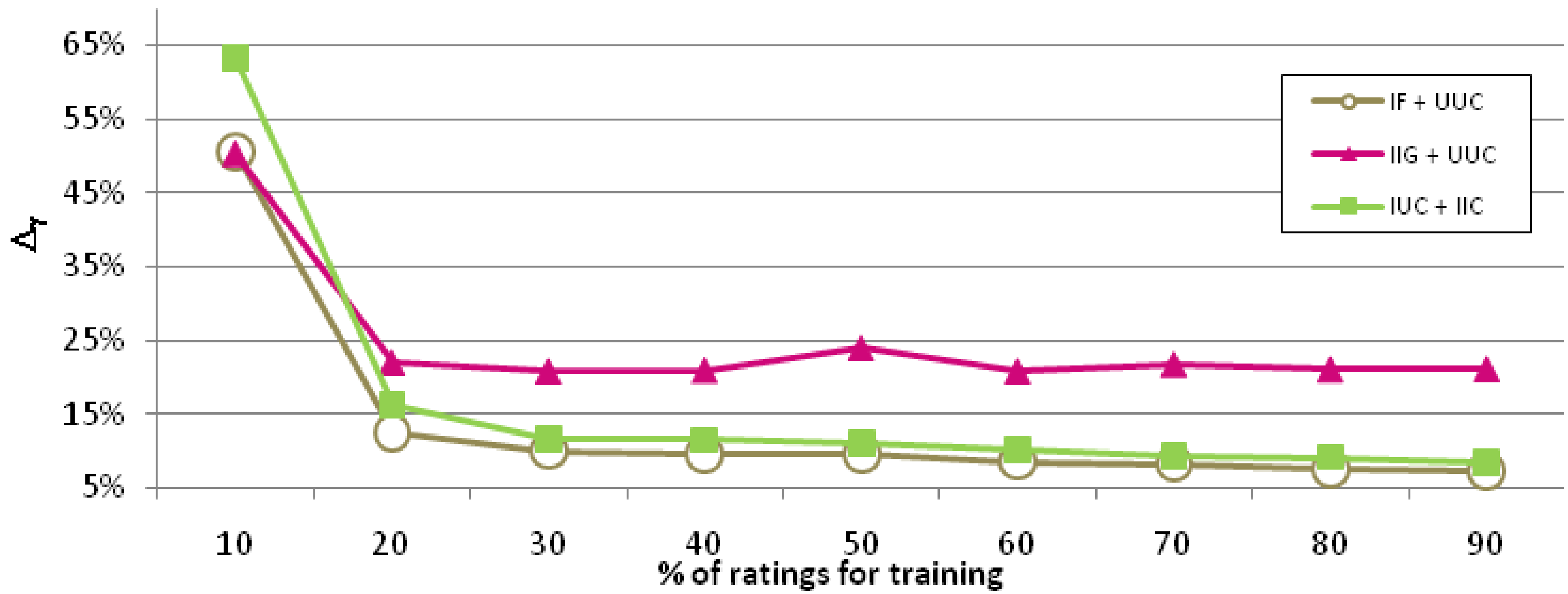
Weighted hybrid recommendation

- MAE is not discriminative enough (in this context)
- New measure: it reflects how better the hybrid combination weights are chosen by dynamic hybrid recommenders compared to static ones

$$\Delta_\gamma = \frac{100}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{\mathrm{MAE}\left(S\left(\alpha_u\left(u\right)\right)\right) - \mathrm{AE}_\gamma\left(u\right)}{\mathrm{MAE}\left(S\left(\alpha_u\left(u\right)\right)\right)}$$

$$\alpha_u = \frac{\alpha_{CB}}{\alpha_{CB} + \alpha_{CF}}$$

$$\gamma = \left(\gamma_{CB}, \gamma_{CF}\right)$$



Weighted hybrid recommendation

- Correlation analysis with respect to average error (for user / item)
  - Correlation with respect to CF

Predictor de eficacia	% de ratings								
	10	20	30	40	50	60	70	80	90
UUC	-0.44	0.07	0.19	0.20	0.18	0.17	0.17	0.16	0.12
IIC	0.21	0.34	0.31	0.27	0.24	0.21	0.18	0.16	0.13

  - Correlation with respect to CB

Predictor de eficacia	% de ratings								
	10	20	30	40	50	60	70	80	90
IIG	-0.09	-0.07	-0.08	-0.08	-0.10	-0.10	-0.11	-0.10	-0.10
IUC	0.17	0.18	0.16	0.15	0.14	0.14	0.12	0.10	0.09
IF	0.06	0.00	N/A	N/A	-0.07	-0.07	-0.01	-0.03	0.08

- Clarity-based predictors: positive correlation both for CB and CF
- IIG and IF are negatively correlated, although not very significative

Conclusions

- Contributions of the work:
  - A formal framework for the introduction of performance predictors in recommender systems
  - Adaptation of query clarity techniques to recommender systems
  - Definition of new performance predictors for recommender systems based on Information Theory
  - Application to two problems:
    - Neighbour weighting in Collaborative Filtering
    - Hybrid weighting
  - Experimental validation of the proposed methods:
    - Performance analysis of combined systems where predictors are introduced for dynamic weighting of subcomponents
    - Analysis of correlation between the predictors and performance metrics
  - Two new performance measures are proposed: NG y Δ_γ

Conclusions

- Future work:
  - Improvement of predictors and definition of new ones (based on JSD or WIG)
  - Comprehensive analysis of predictors (defining different ψ)
  - Creation of specific datasets
  - Large scale experiments
  - Research of performance measures (properties, behaviour)
  - Extension of formal framework
  - Extension to new areas: personalised search, context-based retrieval, metasearch, distributed search

Thank you

Bibliography

- (Amati et al. 2004) Amati, G., Carpineto, C. & Romano, G. (2004), ‘Query difficulty, robustness, and selective application of query expansion’, *Advances in Information Retrieval* pp. 127–137.
- (Aslam & Pavlu 2007) Aslam, J. A. & Pavlu, V. (2007), Query hardness estimation using Jensen-Shannon divergence among multiple scoring functions, *in* ‘ECIR’, pp. 198–209.
- (Carmel et al. 2006) Carmel, D., Yom-Tov, E., Darlow, A. & Pelleg, D. (2006), What makes a query difficult?, *in* ‘SIGIR ’06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval’, ACM, New York, NY, USA, pp. 390–397.
- (Castells et al. 2005) Castells, P., Fernández, M., Vallet, D., Mylonas, P. & Avrithis, Y. (2005), Self-tuning personalized information retrieval in an ontology-based framework, *in* R. Meersman, Z. Tari & P. Herrero, eds, ‘SWWS’05: In proceedings of the 1st International Workshop on Web Semantics’, Vol. 3762, Springer, pp. 977–986.
- (Cronen-Townsend et al. 2002) Cronen-Townsend, S., Zhou, Y. & Croft, B. W. (2002), Predicting query performance, *in* ‘SIGIR ’02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval’, ACM Press, New York, NY, USA, pp. 299–306.
- (Diaz 2007) Diaz, F. (2007), Performance prediction using spatial autocorrelation, *in* ‘SIGIR ’07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval’, ACM, New York, NY, USA, pp. 583–590.

Bibliography

- (Diaz & Jones 2004) Diaz, F. & Jones, R. (2004), Using temporal profiles of queries for precision prediction, *in* ‘SIGIR ’04: Proceedings of the 27th annual international conference on Research and development in information retrieval’, ACM Press, pp. 18–24.
- (He & Ounis 2004) He, B. & Ounis, I. (2004), Inferring query performance using pre-retrieval predictors, *in* ‘String Processing and Information Retrieval, SPIRE 2004’, pp. 43–54.
- (Jensen et al. 2005) Jensen, E. C., Beitzel, S. M., Grossman, D., Frieder, O. & Chowdhury, A. (2005), Predicting query difficulty on the web by learning visual clues, *in* ‘SIGIR ’05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval’, ACM, New York, NY, USA, pp. 615–616.
- (Mothe & Tanguy 2005) Mothe, J. & Tanguy, L. (2005), Linguistic features to predict query difficulty, *in* ‘Predicting Query Difficulty - Methods and Applications, SIGIR 2005’.
- (Yom-Tov et al. 2005) Yom-Tov, E., Fine, S., Carmel, D. & Darlow, A. (2005), Metasearch and federation using query difficulty prediction, *in* ‘Predicting Query Difficulty - Methods and Applications, SIGIR 2005’.(Zhou & Croft 2006) Zhou, Y. & Croft, B. W. (2006), Ranking robustness: a novel framework to predict query performance, *in* ‘CIKM ’06: Proceedings of the 15th ACM international conference on Information and knowledge management’, ACM, New York, NY, USA, pp. 567–574.
- (Zhou & Croft 2007) Zhou, Y. & Croft, B. W. (2007), Query performance prediction in web search environments, *in* ‘SIGIR ’07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval’, ACM, New York, NY, USA, pp. 543–550.