Induction of user preferences

Alan Eckhardt

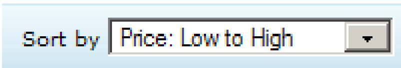
Department of Software Engineering,
Charles University in Prague,
Czech Republic

Outline

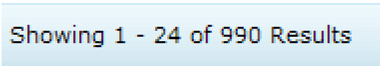
- Motivation
- User model
- Learning the model
- Experiments
- Conclusion

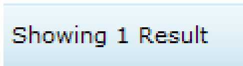
Motivation – current state

- Classical e-shop
  - Strict criteria on attributes
    - Price 50-100$
  - Simple ordering by price, name



  - Many or few objects





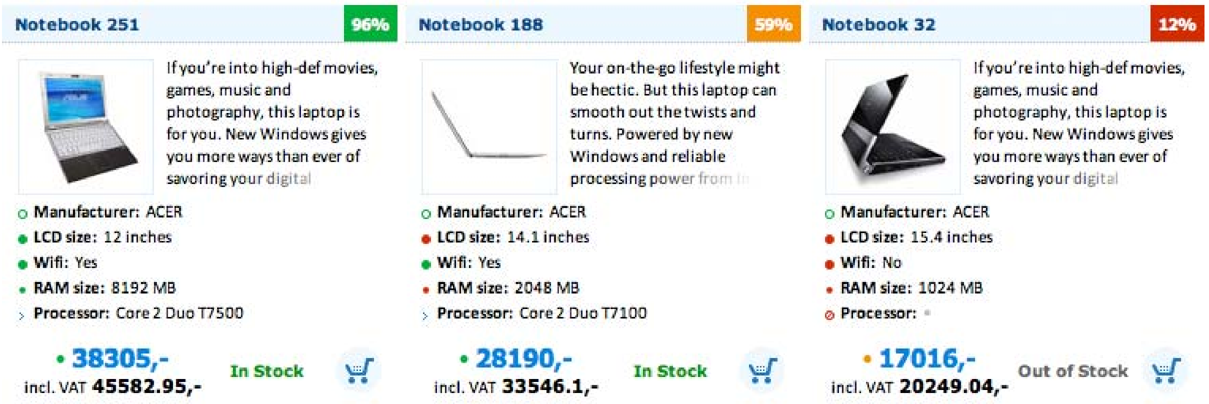
  - Same answer for every user

Motivation – dream state

- Personalization
  - Results reflect the user, not only the query
- More insight for the user
  - Solve „no object“ or „too many objects“ problem
  - *How* good is the notebook?
    - Relaxed criteria
    - More complex ordering

Motivation – preference search

- Ordering
  - Instead of restriction
- Specification of ideal attribute values
  - Rather than acceptable values
- Possible visualization of the result set



Motivation – preference learning

- Let's expect even less from the user
  - Instead of direct specification of ideal values - learning the most preferred values from a less complex user input
  - A few ratings of objects

Toshiba Portege R600

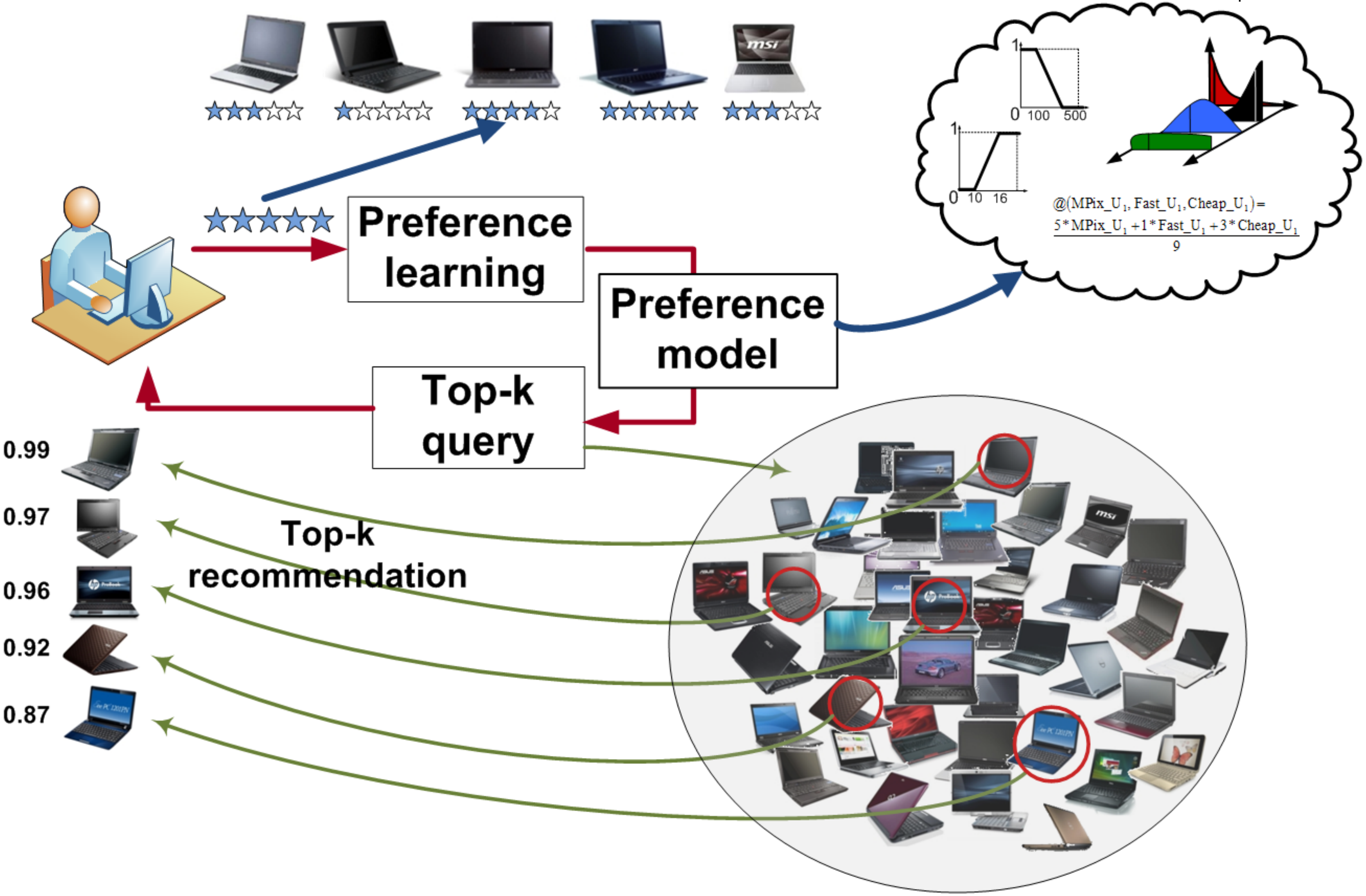
Toshiba's Portege R600 is one of the best ultraportables on the market, if you're willing to pay the price.

Tags: toshiba, portege, r600, ultraportable, laptop

Your rating: ★★★☆☆

- Construction of a general user preference model
  - Each user has his/her own preference model
  - Based on his/her ratings

Motivation – preference learning



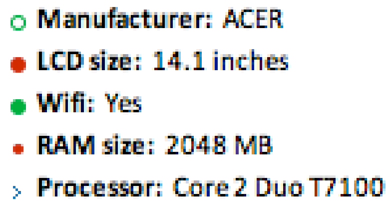
User model

- User model learning is divided into two steps

1. **Local preferences** - normalization of the attribute values of notebooks to their preference degrees

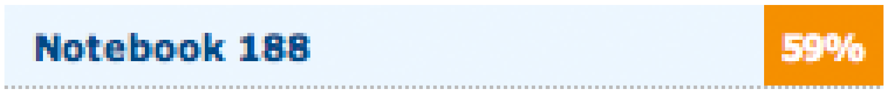
   $f_i : D_{A_i} \rightarrow [0,1]$

   Transforms the space ΠD_{A_i} into $[0,1]^N$



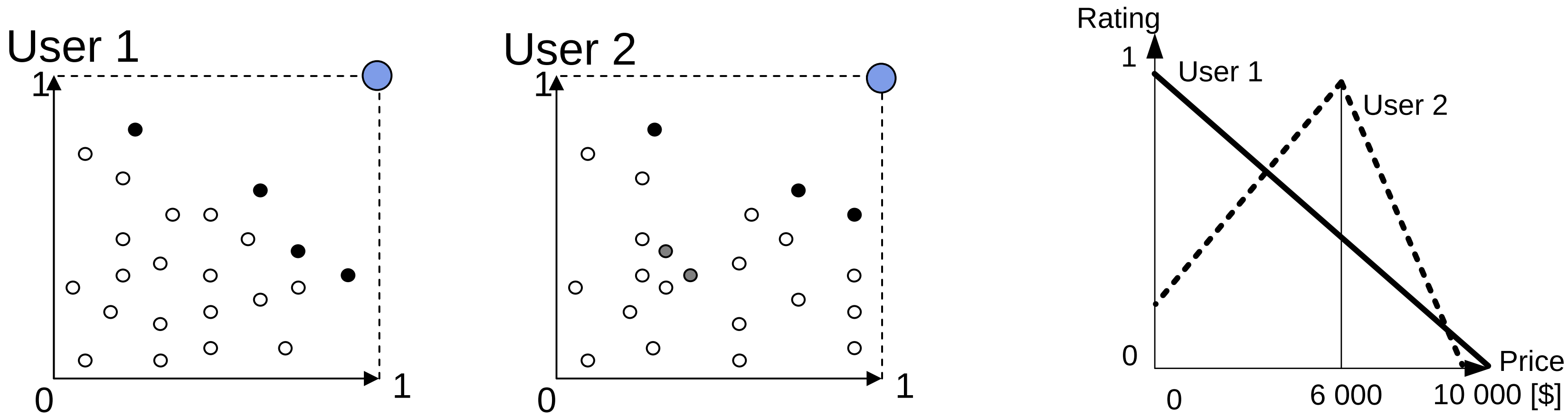
2. **Global preferences** - aggregation of preference degrees of attribute values into the predicted rating

   $@ : [0,1]^N \rightarrow [0,1]$



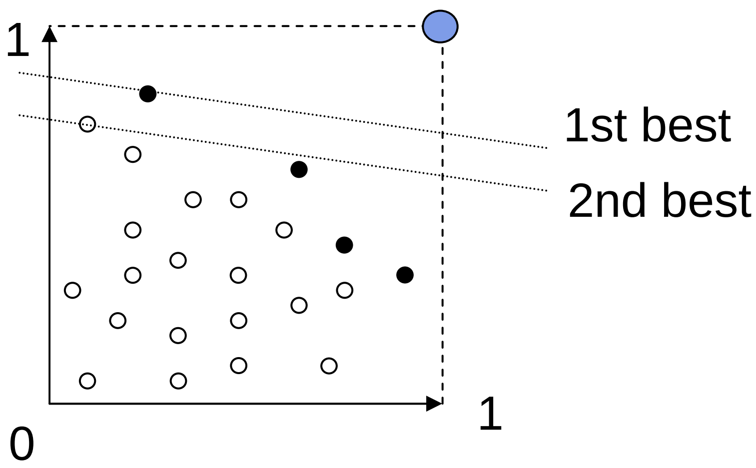
User model

- Local preferences
  - Transform the space ΠD_{A_i} to monotone space $[0,1]^N$
    - ● = (1,…,1) is the best object
    - Allow direct comparison of objects



User model

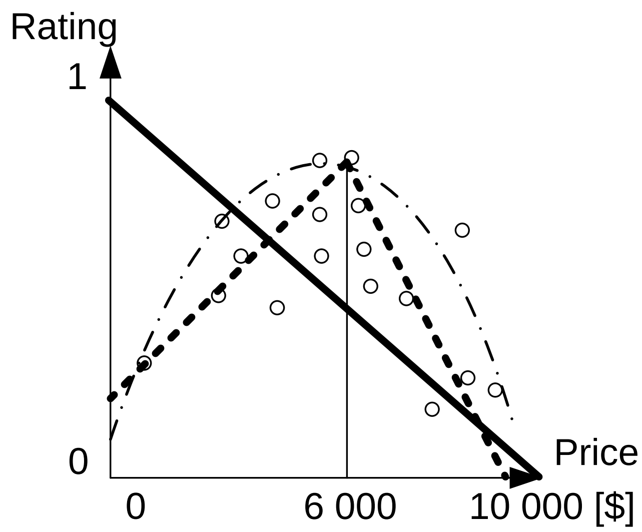
- Global preferences $@:[0,1]^N \rightarrow [0,1]$
  - Order all objects according to overall preference
  - Allow recommendation of top-k objects



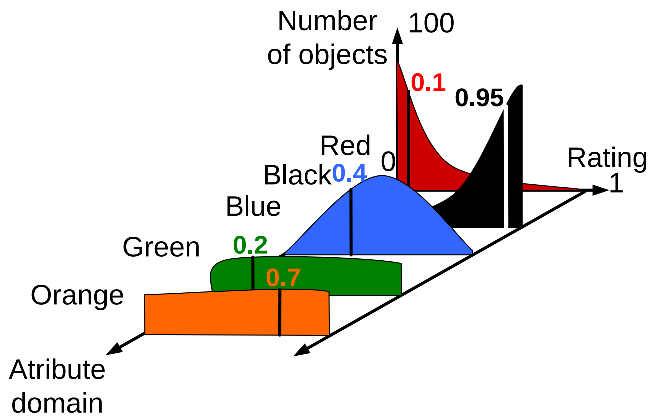
$$@\left(\mathrm{RAM\_U_1}, \mathrm{CPU\_U_1}, \mathrm{Price\_U_1}\right) = \frac{5 * \mathrm{R}AM\_\mathrm{U}_1 + 1 * \mathrm{CPU\_U}_1 + 3 * \mathrm{Price\_U}_1}{9}$$

Learning user model

- Regression for numerical attributes

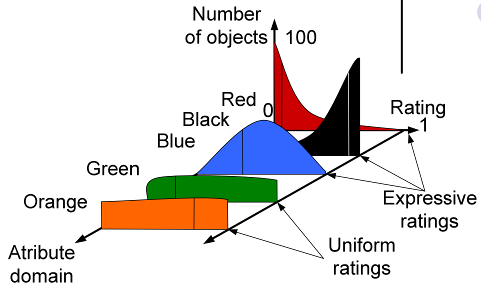


- Average rating for nominal attributes



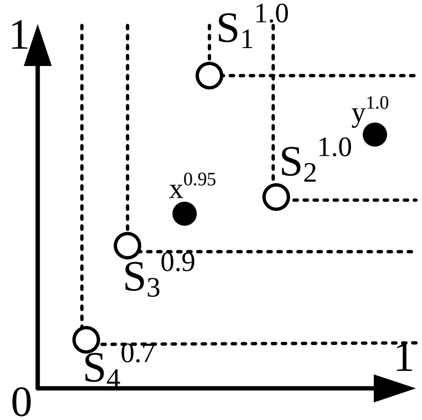
Learning user model

- *Statistical*
  - Learned weights for weighted average
  - Using distribution of ratings



$$@(\mathrm{RAM\_U}_1, \mathrm{CPU\_U}_1, \mathrm{Price\_U}_1) = \frac{5 * \mathrm{RAM\_U}_1 + 1 * \mathrm{CPU\_U}_1 + 3 * \mathrm{Price\_U}_1}{9}$$

- *Instances*
  - Uses objects from training set for estimation of rating



Specific problems to preference learning

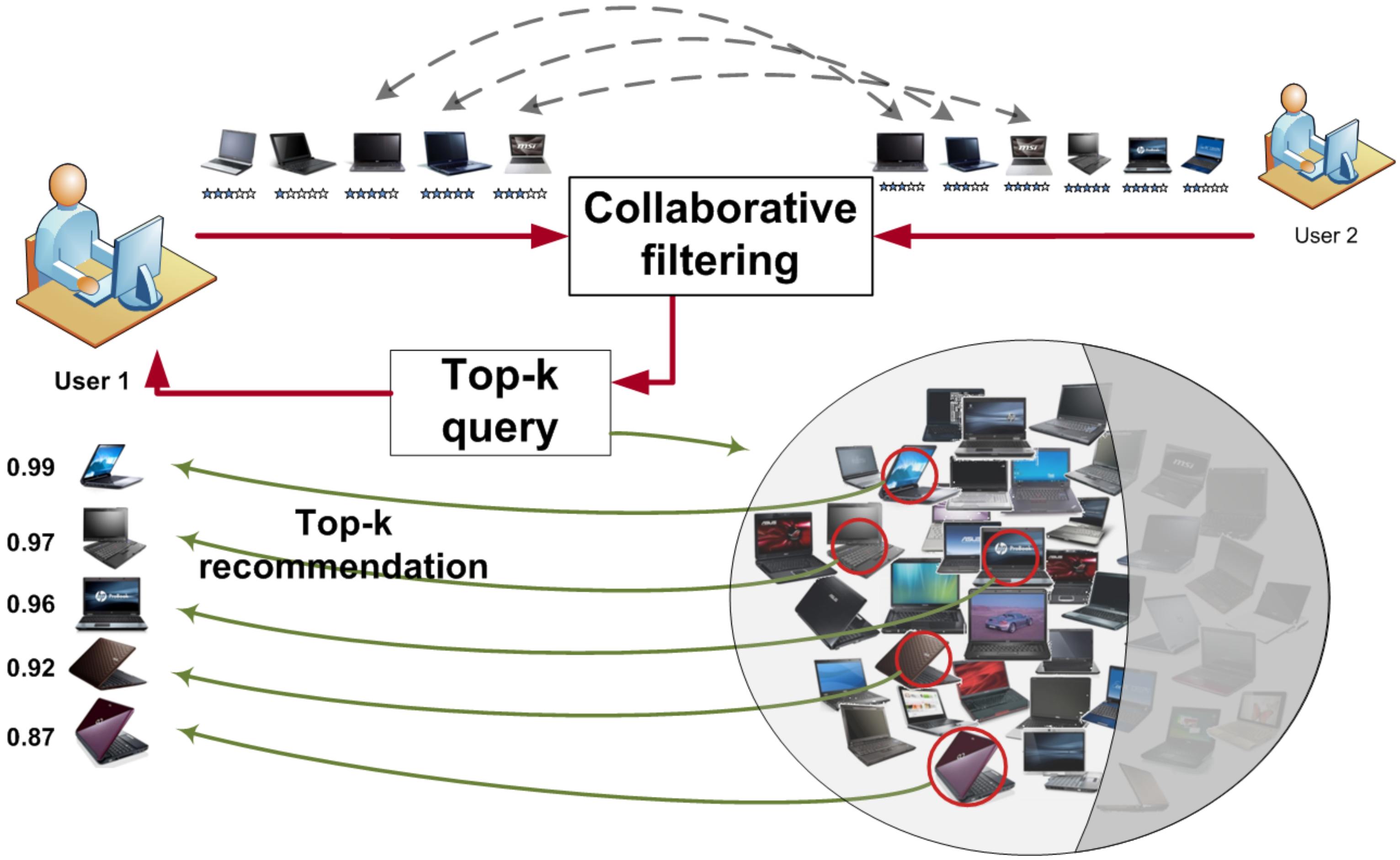
- Small training set
  - Users do not want to invest too much effort
- Different error measures than “standard” machine learning
  - Correct ordering is important
  - Better rated objects are more important
- Transparent preference model needed
  - For using the information in user interface
    - Why is the object recommended by the system?

Specific problems to preference learning

- Error measures
  - WRMSE
    - $\sqrt{\sum_{o \in X} r(o)(\hat{r}(o) - r(o))^2 / \sum_{o \in X} r(o)}$
  - WTau coefficient
    - Compares two ordered lists and emphasizes better rated objects
  - Top-k score
    - Percentage of correct objects in top-k (without order)

Collaborative filtering

- Similarity of users based on ratings of common rated objects
  - Only users with common rated objects can be evaluated



Combination with other methods

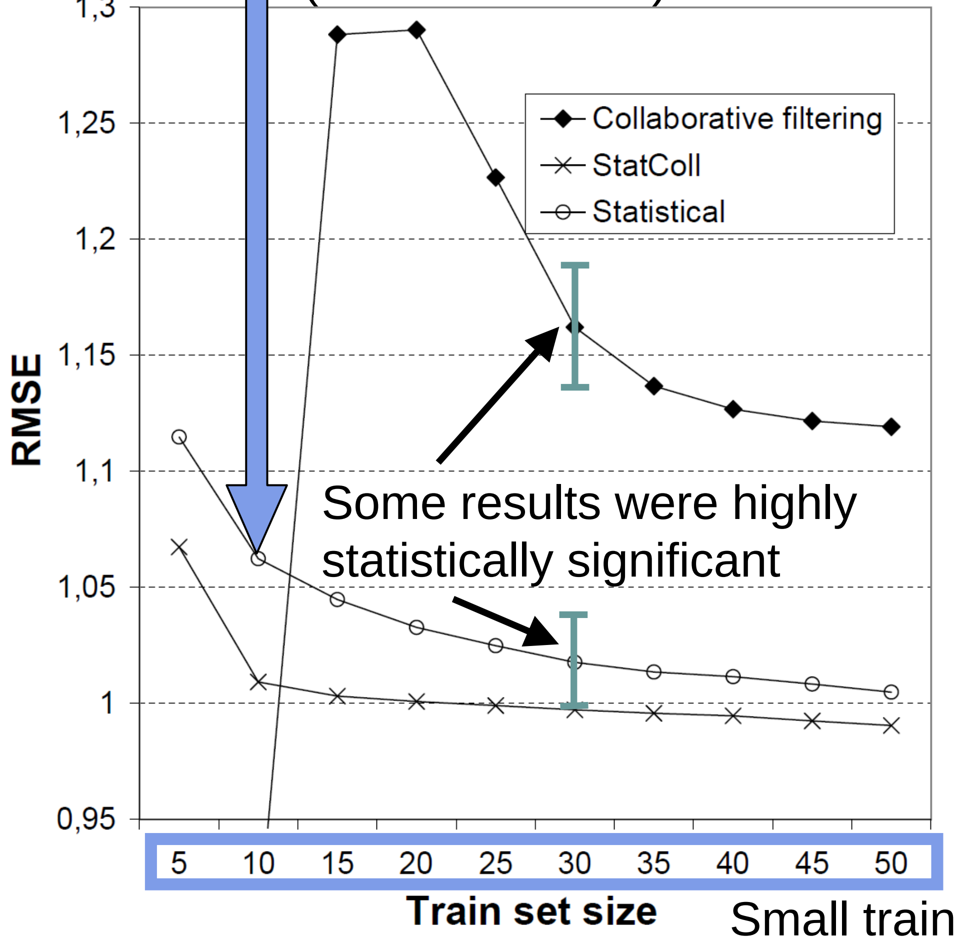
- Similarity of user preference models for collaborative filtering
  - StatColl
    - Similarity of local and global preferences
      - All users can be evaluated, every user has preference model
    - In normal CF, similarity based on ratings
      - Only users with common rated objects can be evaluated

Experiment results

- **Enhancing Collaborative filtering**
  - *StatColl* uses similarity of user models (Statistical)
    - Enables computing the similarity for all users
  - StatColl statistically significantly better
  - Netflix dataset + IMDb dataset

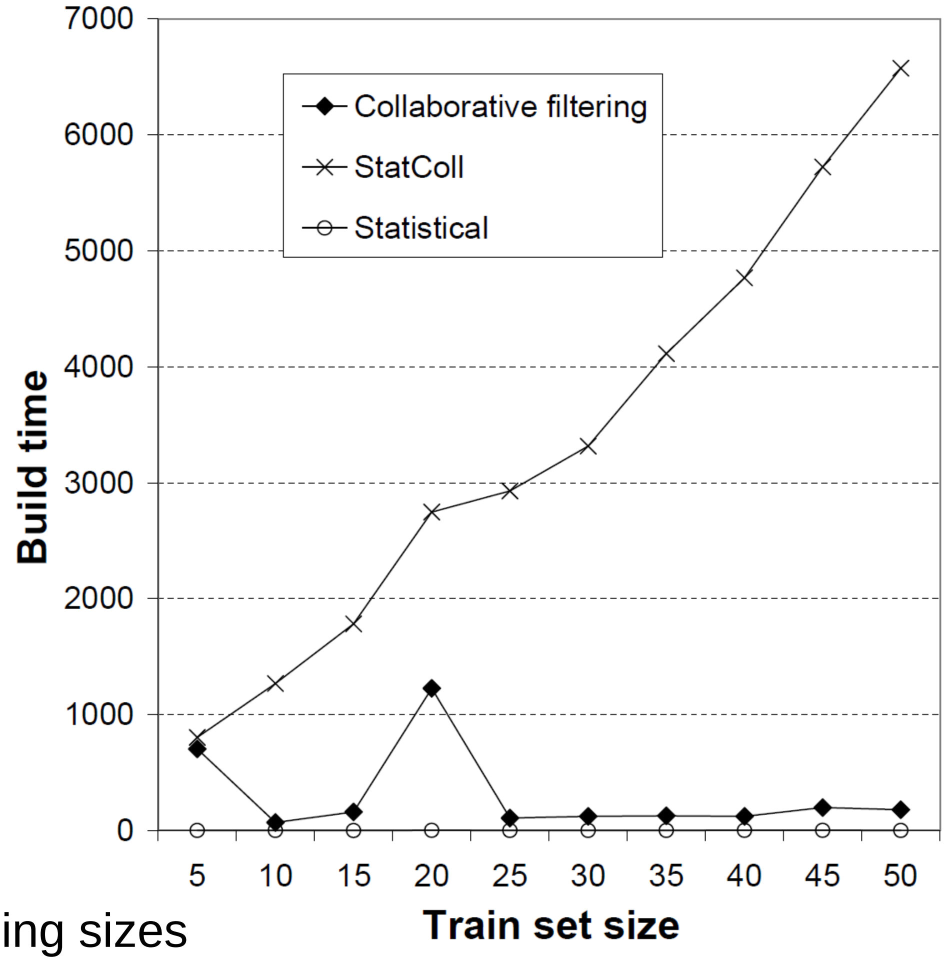
Experiment results

One point in the graph represents the average of all runs and all datasets / users (10000 numbers)

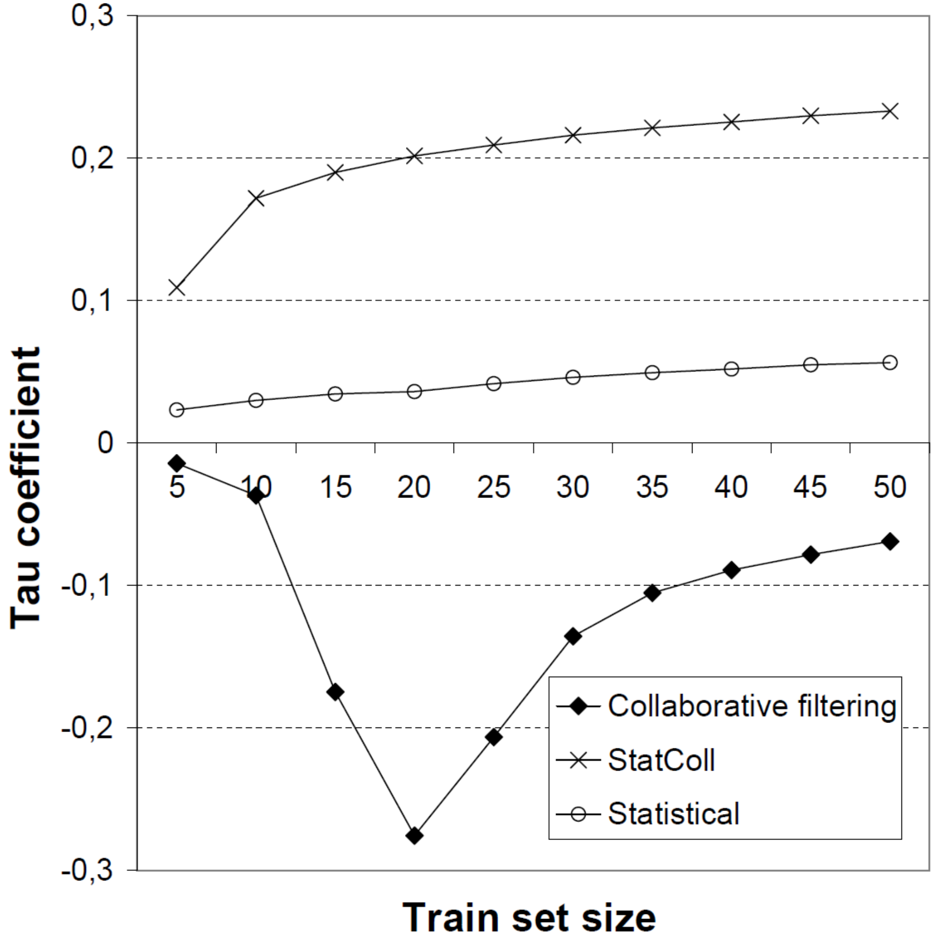


Some results were highly statistically significant

Small training sizes



Experiment results



Tau coefficient is negative for CF, because it has a lot of unpredicted objects

Experiment realization

- PrefWork was used for performing the experiments
  - Easy configuration, alteration, repetition of experiments
  - New methods, datasets, error measures, testing setting, integration of other methods

Conclusion

- Proposal of various methods for preference learning
  - New approaches specific for preferences, addressing problems of preference learning, implemented, ready to use in a web shop - a transparent preference model
- Experimental verifications of several new approaches
  - Statistically significant improvement of various methods
  - Enhancement of existing methods
- PrefWork for performing experiments

Future work

- Extending the e-shop with even more preferences
  - Vaclav - Java, Kolomicenko - PHP
  - New user interfaces
  - New paradigm for preference use
- New methods for learning for PrefWork
  - From user behavior
  - Learning of preference relations