Personalized Transaction Kernels for Recommendation using MCTS

Maryam Tavakol, Tobias Joppen,
Ulf Brefeld, and Johannes Fürnkranz

Kassel, 25.09.2019

Outline

- Introduction and motivation
- Personalized transaction kernels
- Informed sampling strategies
- Empirical study
- Summary and Conclusions

Recommender Systems

- A user views the following item:

Goal: recommend an item(s) that is relevant to the user

Task: capture/understand the interests of users

Traditional Approaches

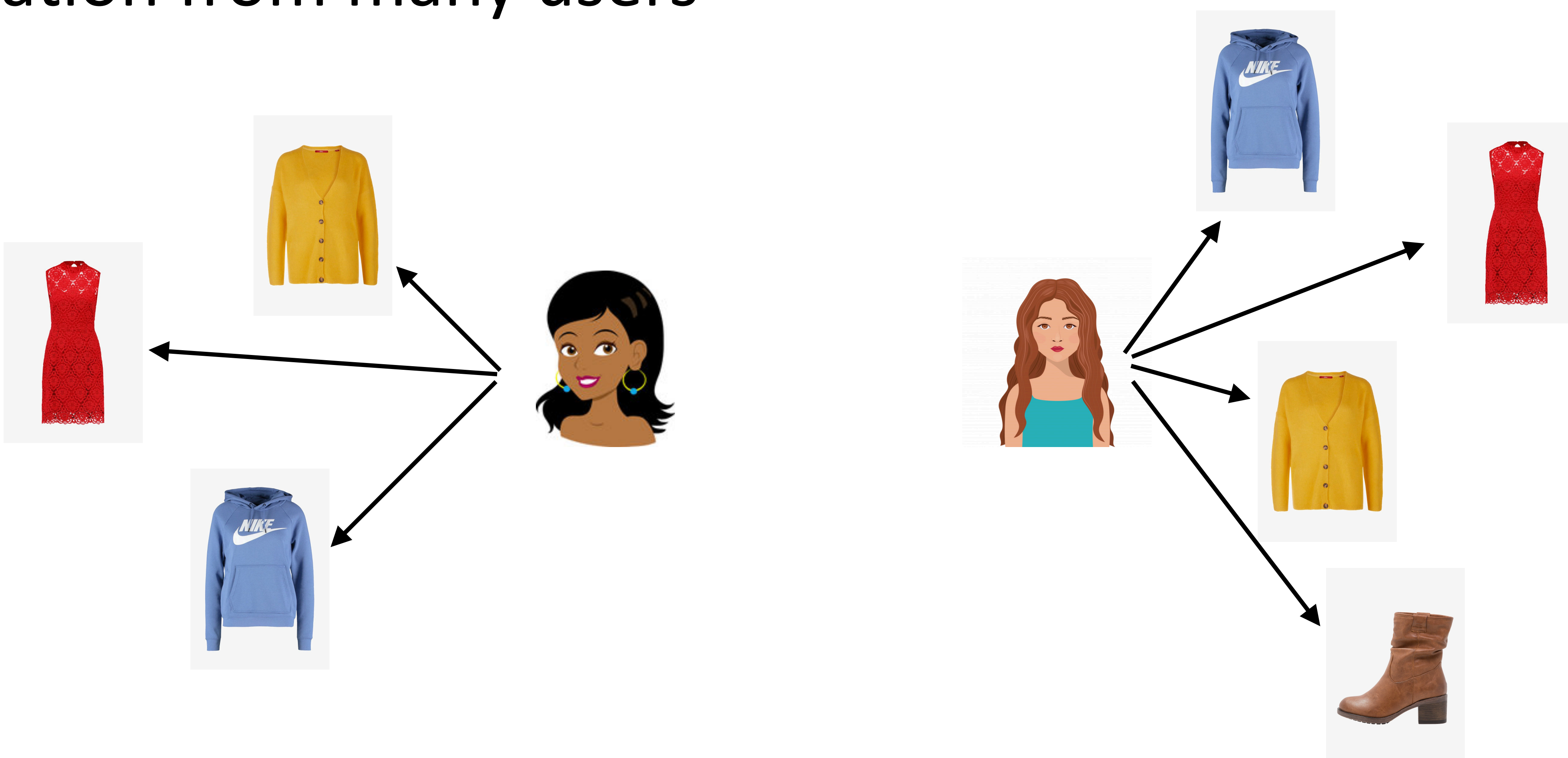
Collaborative filtering (Resnick et al., 1994): based on collecting information from many users

Traditional Approaches

Collaborative filtering (Resnick et al., 1994): based on collecting information from many users

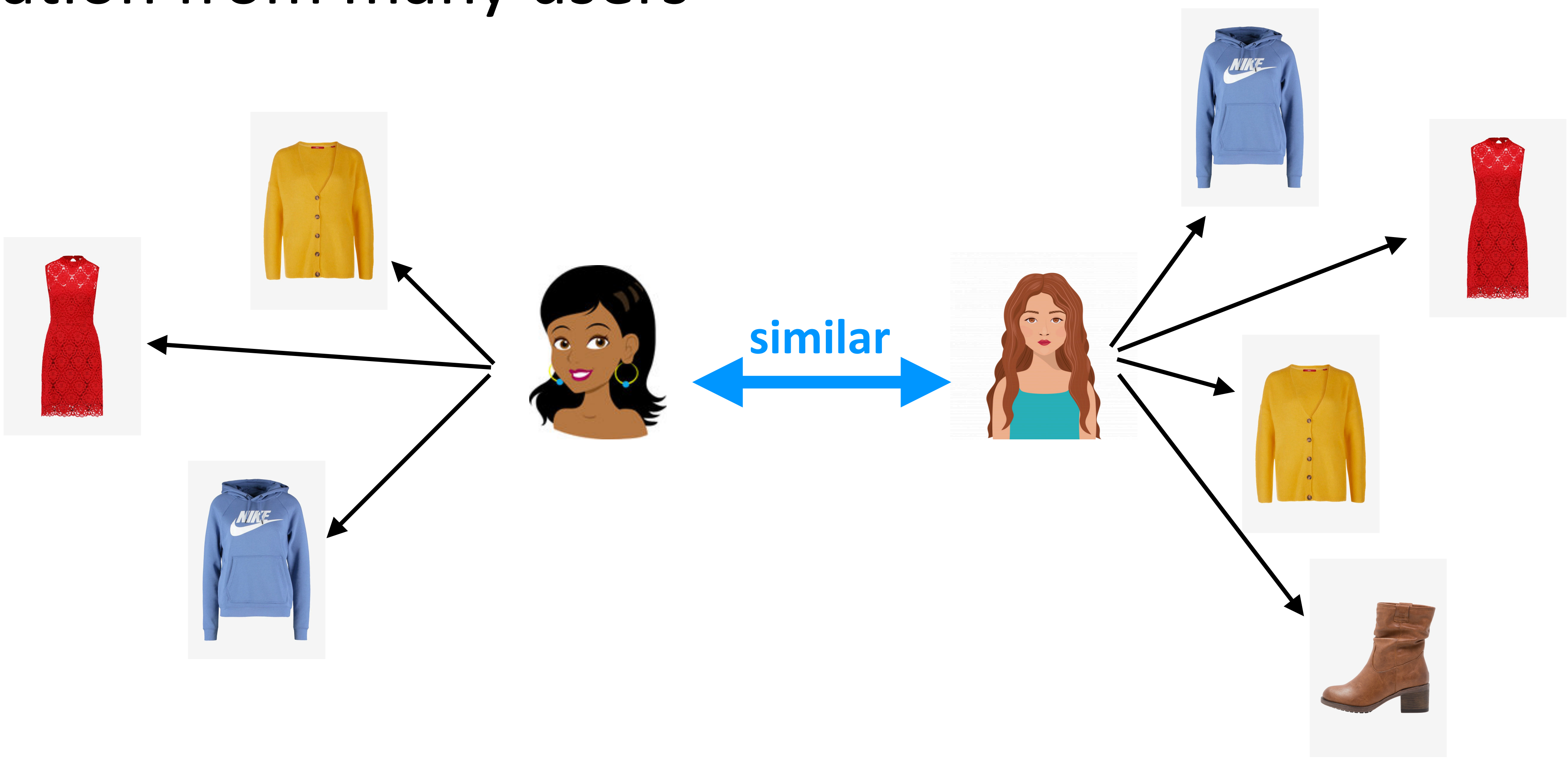
Traditional Approaches

Collaborative filtering (Resnick et al., 1994): based on collecting information from many users



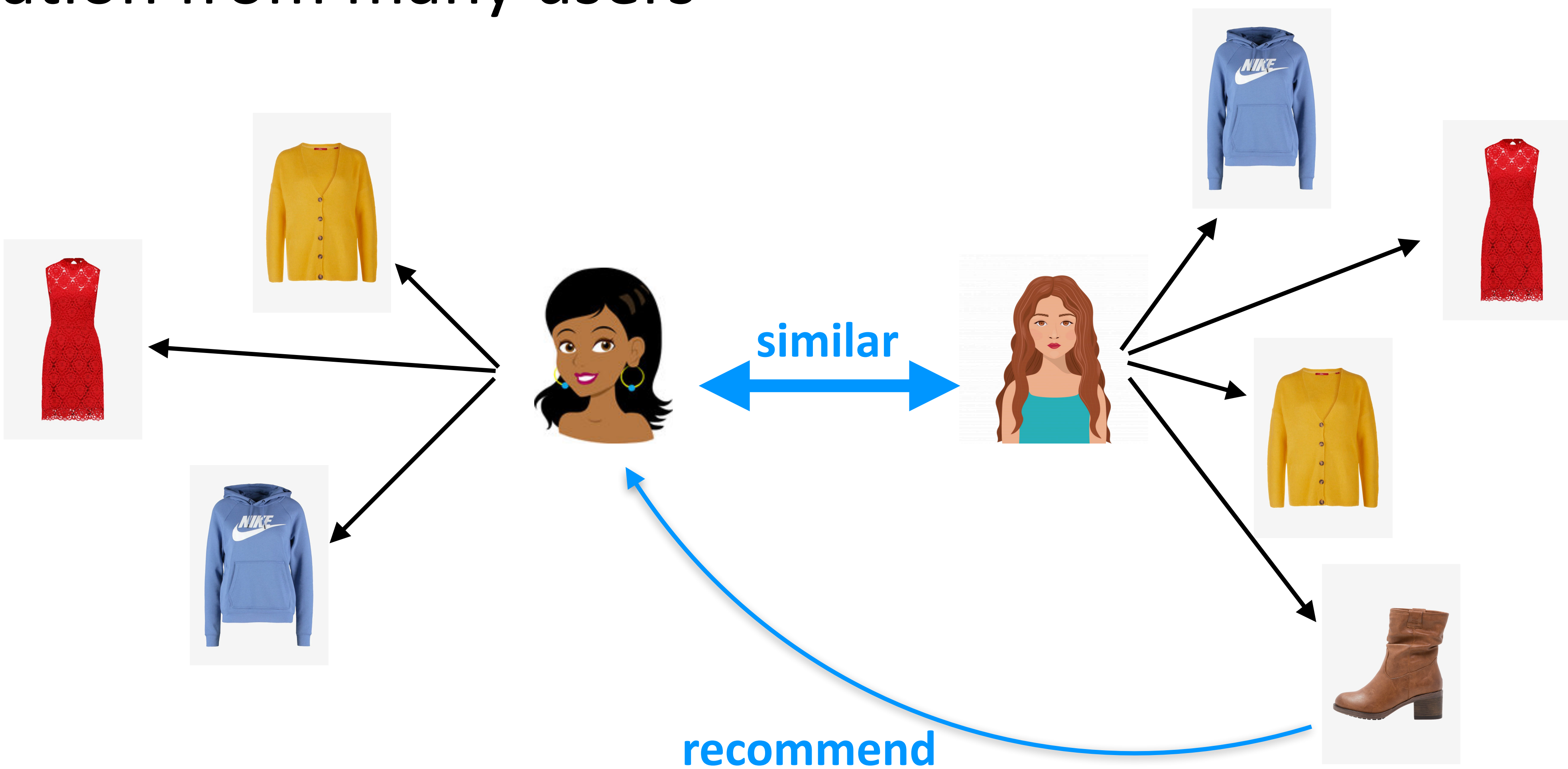
Traditional Approaches

Collaborative filtering (Resnick et al., 1994): based on collecting information from many users



Traditional Approaches

Collaborative filtering (Resnick et al., 1994): based on collecting information from many users



Shortcomings

Shortcomings

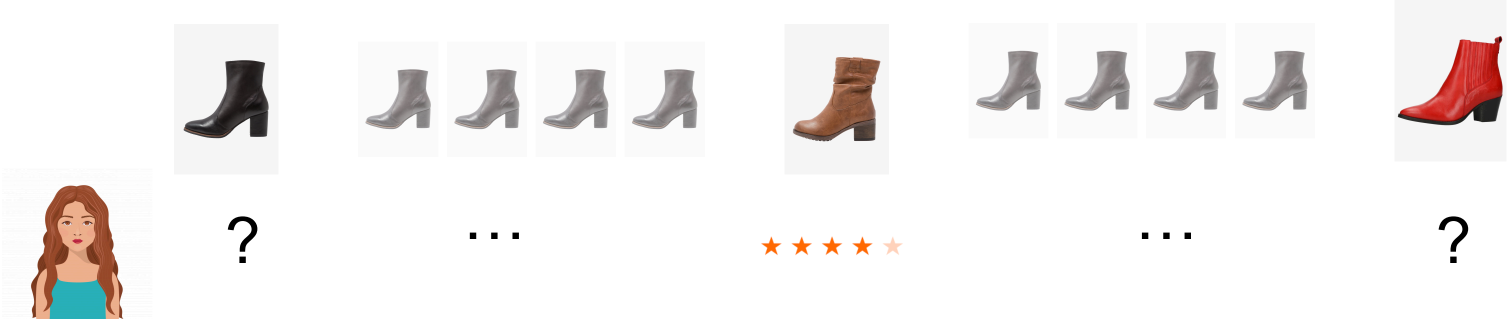
- Operates with explicit feedback

Shortcomings

- Operates with explicit feedback
- Computationally expensive for large-scale problems

Shortcomings

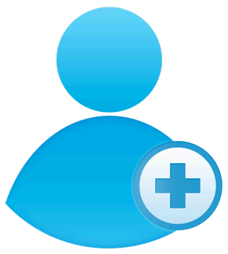
- Operates with explicit feedback
- Computationally expensive for large-scale problems
- Does not cope with data sparsity



Shortcomings

- Operates with explicit feedback
- Computationally expensive for large-scale problems
- Does not cope with data sparsity
- Fails in cold-start problem

?



?

Filling the Gap

- Explicit feedback → **preference-based feedback**
- Computational complexity:
  - Users (personalization) → **efficient multi-task learning**
  - Items (online recommendation) → **informed search**
- Data sparsity → **contextual model**
- Cold-start problem → **contextual model**

Qualitative Feedback

A user views the following item:

Qualitative Feedback

A user views the following item:

Qualitative Feedback

A user views the following item:

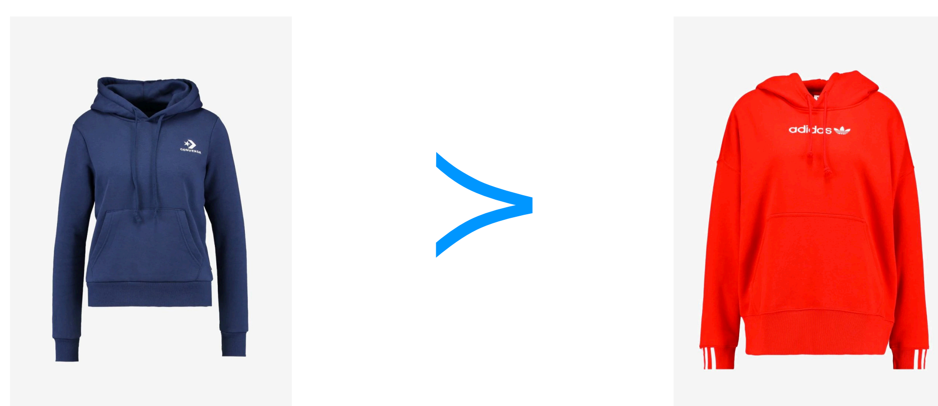
recommended items:

Qualitative Feedback

A user views the following item:

Qualitative Feedback

A user views the following item:



Main Contributions

- An **efficient** preference model is proposed for recommendation with qualitative feedback

Main Contributions

- An **efficient** preference model is proposed for recommendation with qualitative feedback

contextual model: tensor product

Main Contributions

- An **efficient** preference model is proposed for recommendation with qualitative feedback

contextual model: tensor product

↓

personalization: user-specific hashing functions

Main Contributions

- An **efficient** preference model is proposed for recommendation with qualitative feedback

contextual model: tensor product

personalization: user-specific hashing functions

personalized transaction kernel

Main Contributions

- An **efficient** preference model is proposed for recommendation with qualitative feedback

contextual model: tensor product

↓

personalization: user-specific hashing functions

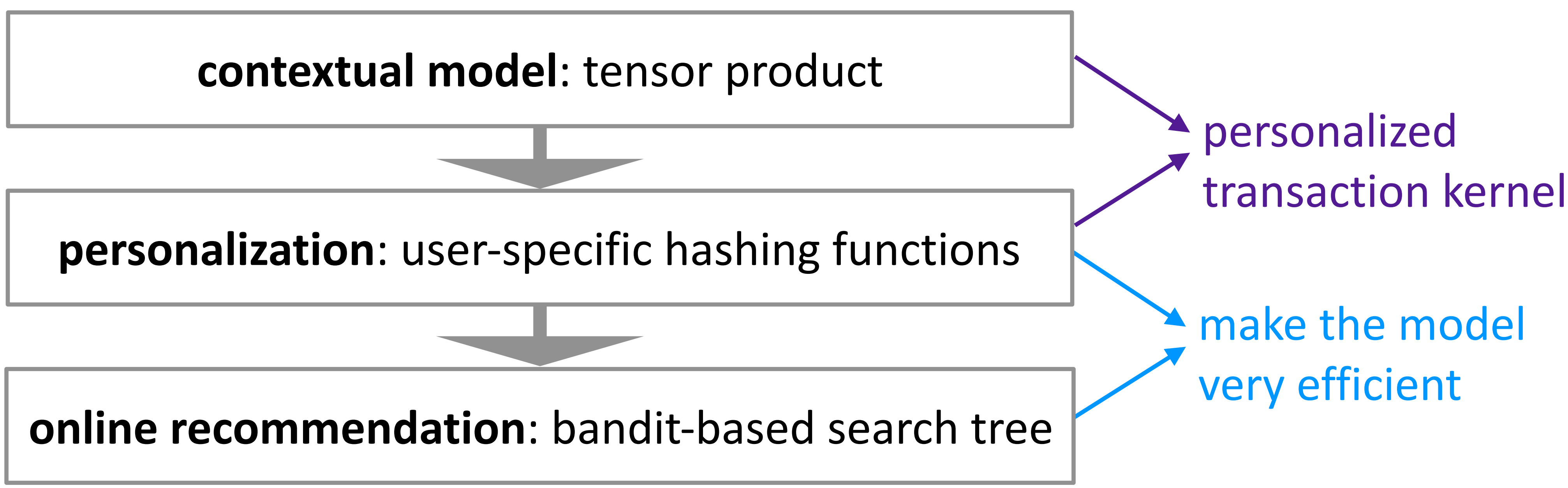
↓

online recommendation: bandit-based search tree

personalized transaction kernel

Main Contributions

- An **efficient** preference model is proposed for recommendation with qualitative feedback



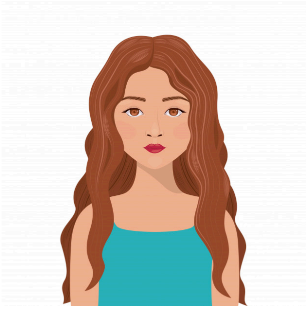
Outline

- Introduction and motivation
- Personalized transaction kernels
- Informed sampling strategies
- Empirical study
- Summary and Conclusions

The Preference Model

The Preference Model

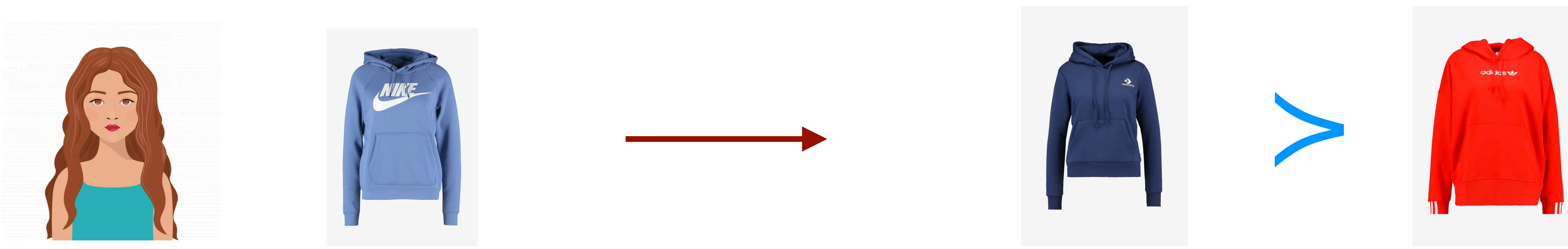
user



context

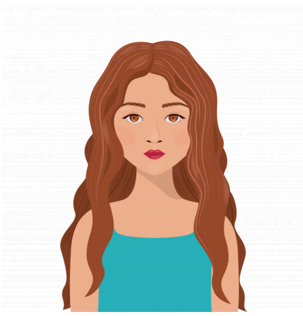
The Preference Model

user | context | pairwise preference



The Preference Model

user



u

context

$\mathbf{s}_t$

pairwise preference

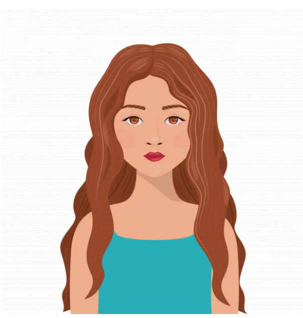
 $\succ$ 

$\mathbf{x}_i \succ \mathbf{x}_j$

$$\boxed{f(\mathbf{x}_i | u, \mathbf{s}_t) \succ f(\mathbf{x}_j | u, \mathbf{s}_t)}$$

The Preference Model

user



u

context

$\mathbf{s}_t$

pairwise preference

$\succ$

$\mathbf{x}_i \succ \mathbf{x}_j$

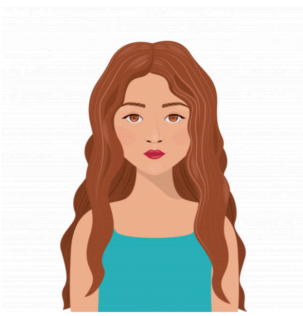
$$\boxed{f(\mathbf{x}_i | u, \mathbf{s}_t) \succ f(\mathbf{x}_j | u, \mathbf{s}_t)}$$

using standard SVM method:
(Joachims, 2002)

$$\mathbf{w}^\top \Big(\phi(\mathbf{x}_i | u, \mathbf{s}_t) - \phi(\mathbf{x}_j | u, \mathbf{s}_t) \Big) \geq 0$$

The Preference Model

user



u

context

$\mathbf{s}_t$

pairwise preference

$\succ$

$\mathbf{x}_i \succ \mathbf{x}_j$

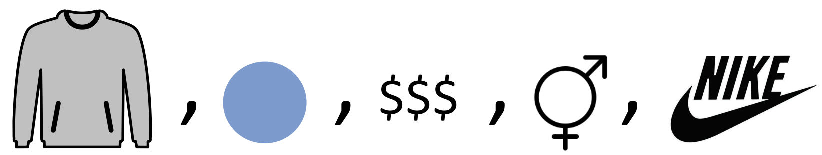
$$\boxed{f(\mathbf{x}_i | u, \mathbf{s}_t) \succ f(\mathbf{x}_j | u, \mathbf{s}_t)}$$

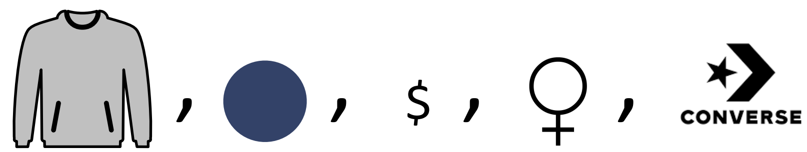
using standard SVM method:
(Joachims, 2002)

$$\mathbf{w}^\top \Big(\phi(\mathbf{x}_i | u, \mathbf{s}_t) - \phi(\mathbf{x}_j | u, \mathbf{s}_t) \Big) \geq 0$$

kernel function

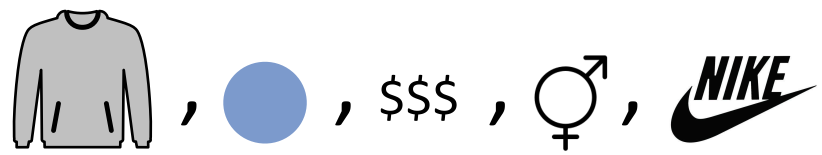
Tensor Kernels

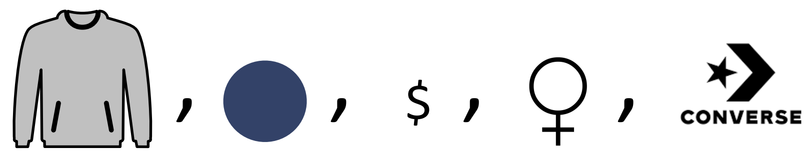
- context: 

- item: 

- context ⊗ item: 

Tensor Kernels

- context: 

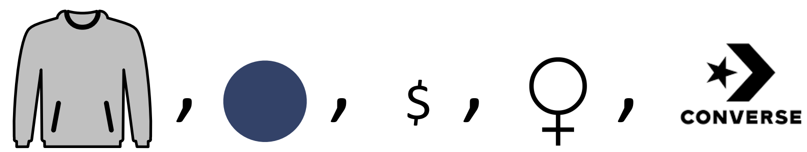
- item: 

- context ⊗ item: 

- Now assume the existence of *m* data sources:

Tensor Kernels

- context: 

- item: 

- context $\otimes$ item: 

- Now assume the existence of *m* data sources:

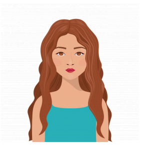
$$\left\langle \{\mathbf{x}^1 \otimes \ldots \otimes \mathbf{x}^m\}, \{\mathbf{z}^1 \otimes \ldots \otimes \mathbf{z}^m\} \right\rangle = \prod_{k=1}^{m} \langle \mathbf{x}^k, \mathbf{z}^k \rangle$$

Personalized Kernels

- Using different hashing functions for users:
  - every user has her own specific representation
  - but in a joint feature space

Personalized Kernels

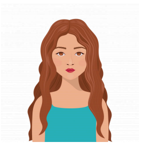
- Using different hashing functions for users:
  - every user has her own specific representation
  - but in a joint feature space

 $\otimes$ context $\otimes$ item

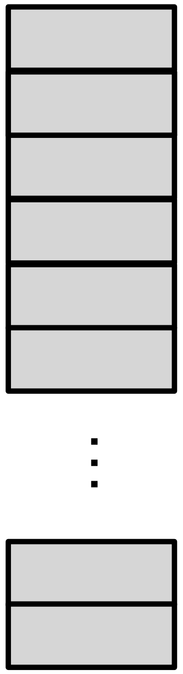
 $\otimes$ context $\otimes$ item

Personalized Kernels

- Using different hashing functions for users:
  - every user has her own specific representation
  - but in a joint feature space

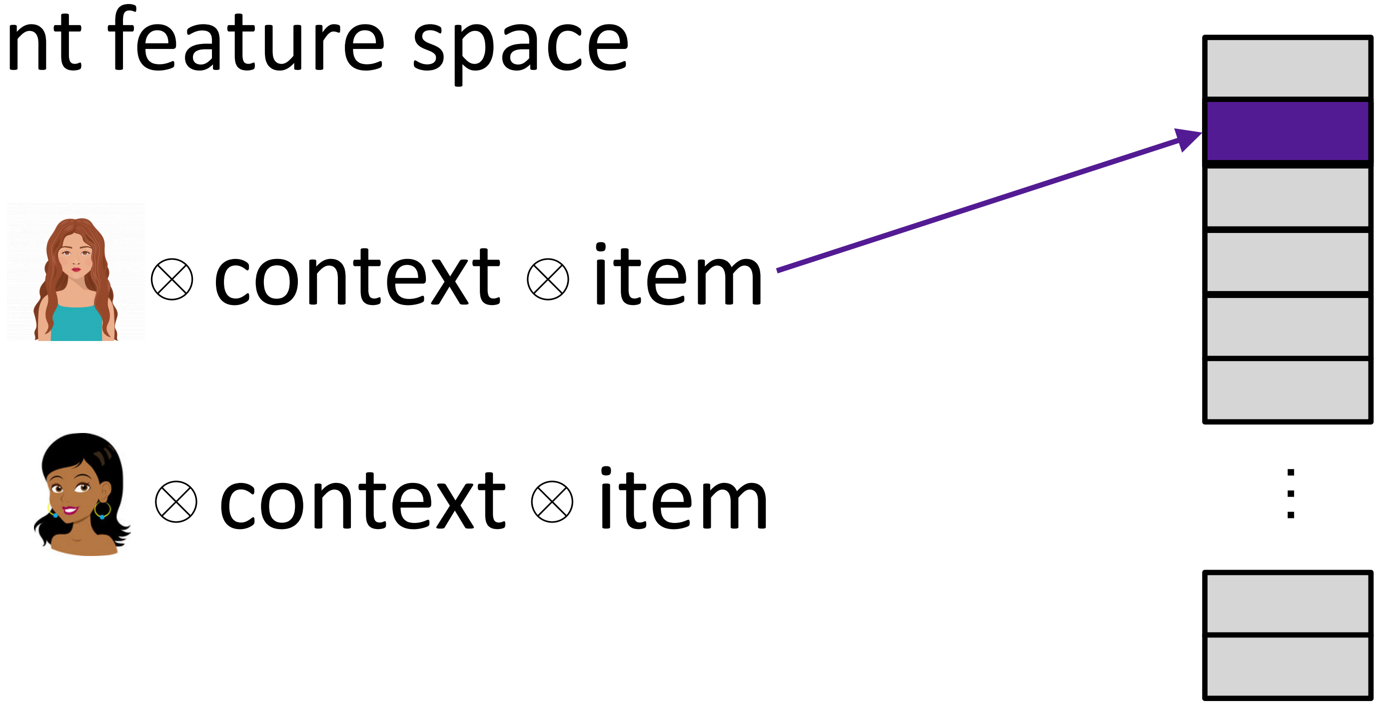
 $\otimes$ context $\otimes$ item

 $\otimes$ context $\otimes$ item



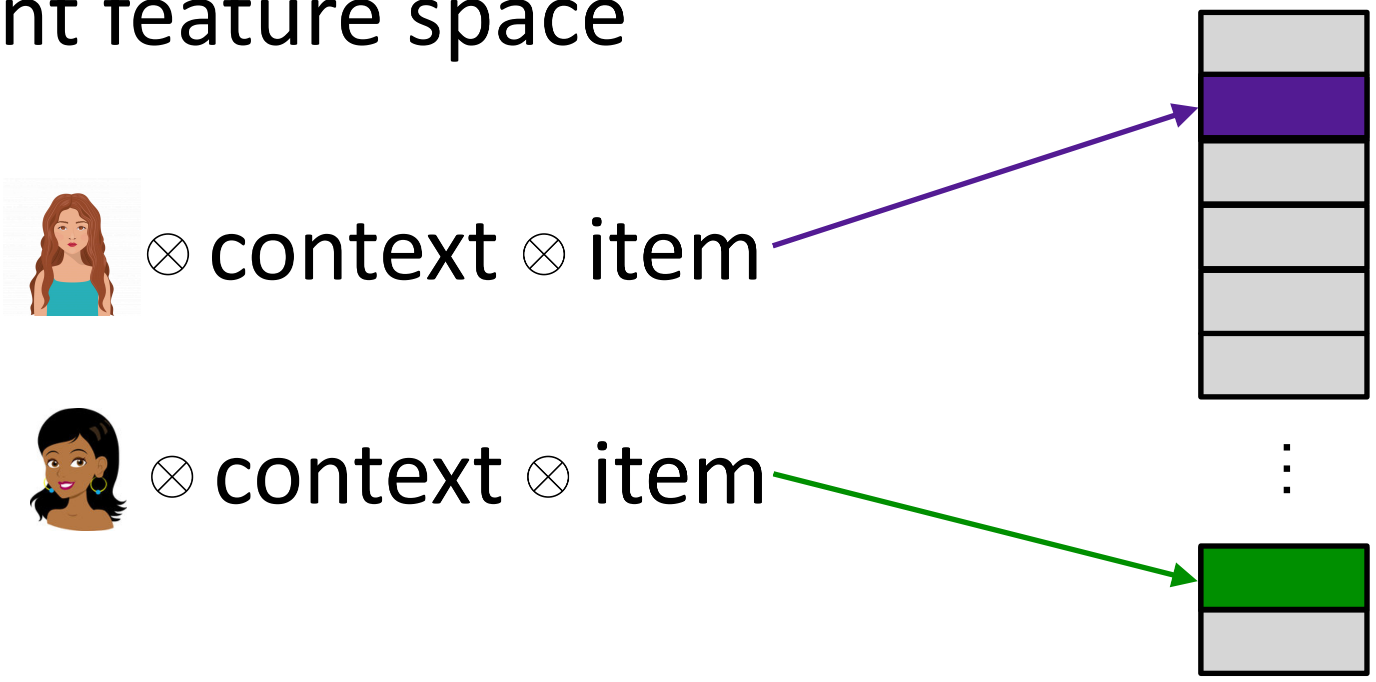
Personalized Kernels

- Using different hashing functions for users:
  - every user has her own specific representation
  - but in a joint feature space



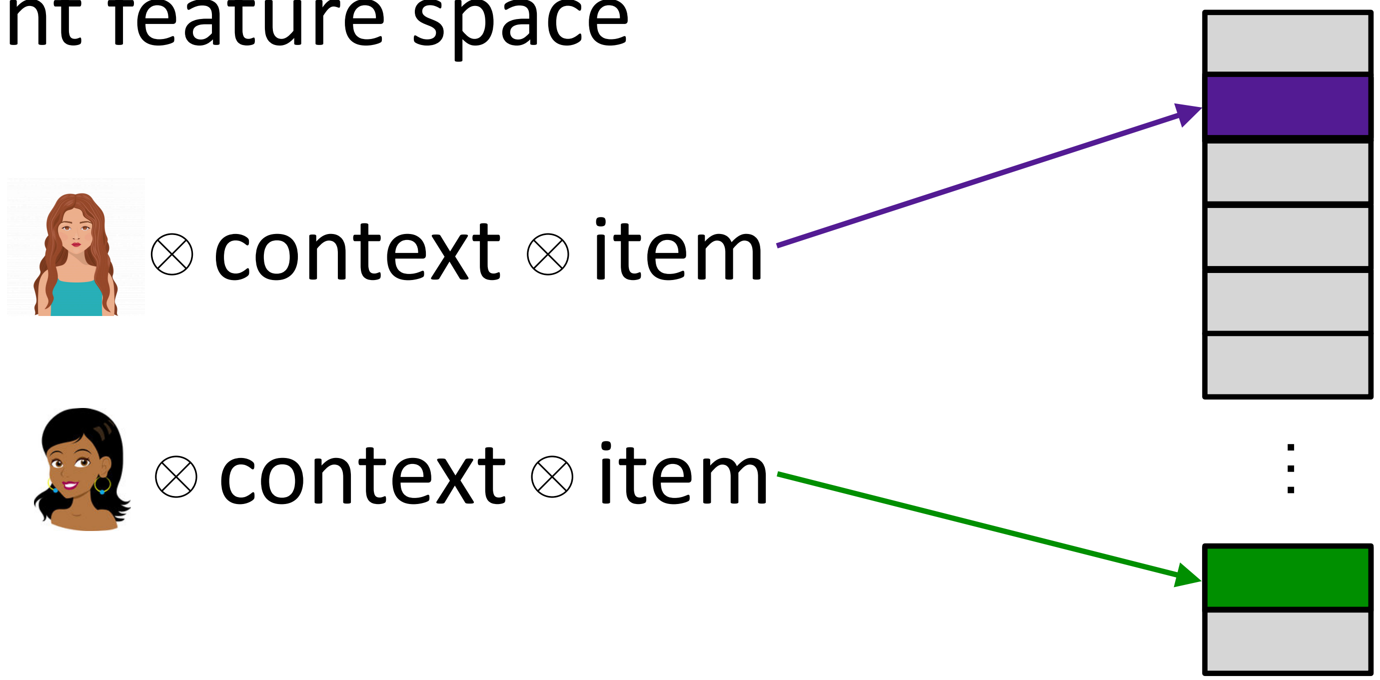
Personalized Kernels

- Using different hashing functions for users:
  - every user has her own specific representation
  - but in a joint feature space



Personalized Kernels

- Using different hashing functions for users:
  - every user has her own specific representation
  - but in a joint feature space



The interaction between random subspaces is negligible with high probability

(Weinberger, 2009)

Preference-based Transaction Kernels

- A **collective** term for all users is added for cold-start scenarios

Preference-based Transaction Kernels

- A **collective** term for all users is added for cold-start scenarios

$$k^{pc} := \underbrace{\text{hash}(\text{user} \otimes \text{context} \otimes \text{item})}_{\text{personalized kernel}} + \underbrace{\text{hash}(\text{context} \otimes \text{item})}_{\text{collective kernel}}$$

Preference-based Transaction Kernels

- A **collective** term for all users is added for cold-start scenarios

$$k^{pc} := \underbrace{\mathrm{hash}(\mathrm{user} \otimes \mathrm{context} \otimes \mathrm{item})}_{\text{personalized kernel}} + \underbrace{\mathrm{hash}(\mathrm{context} \otimes \mathrm{item})}_{\text{collective kernel}}$$

shared for all users

Preference-based Transaction Kernels

- A **collective** term for all users is added for cold-start scenarios

$$k^{pc} := \underbrace{\mathrm{hash}(\mathrm{user} \otimes \mathrm{context} \otimes \mathrm{item})}_{\text{personalized kernel}} + \underbrace{\mathrm{hash}(\mathrm{context} \otimes \mathrm{item})}_{\text{collective kernel}}$$

shared for all users

$$k(\mathbf{x}_i \succ \mathbf{x}_i', \mathbf{x}_j \succ \mathbf{x}_j') =$$

$$k^{pc}(\mathbf{x}_i, \mathbf{x}_j) - k^{pc}(\mathbf{x}_i, \mathbf{x}_j') - k^{pc}(\mathbf{x}_i', \mathbf{x}_j) + k^{pc}(\mathbf{x}_i', \mathbf{x}_j')$$

Outline

- Introduction and motivation
- Personalized transaction kernels
- Informed sampling strategies
- Empirical study
- Summary and Conclusions

Online Recommendation

- How to find the *best* product to recommend?
- Given the preference model:
  - enumerate over all available items
  - compute their utility for a given user and context
  - return the item(s) that maximizes the utility

Online Recommendation

- How to find the *best* product to recommend?
- Given the preference model:
  - enumerate over all available items
  - compute their utility for a given user and context
  - return the item(s) that maximizes the utility

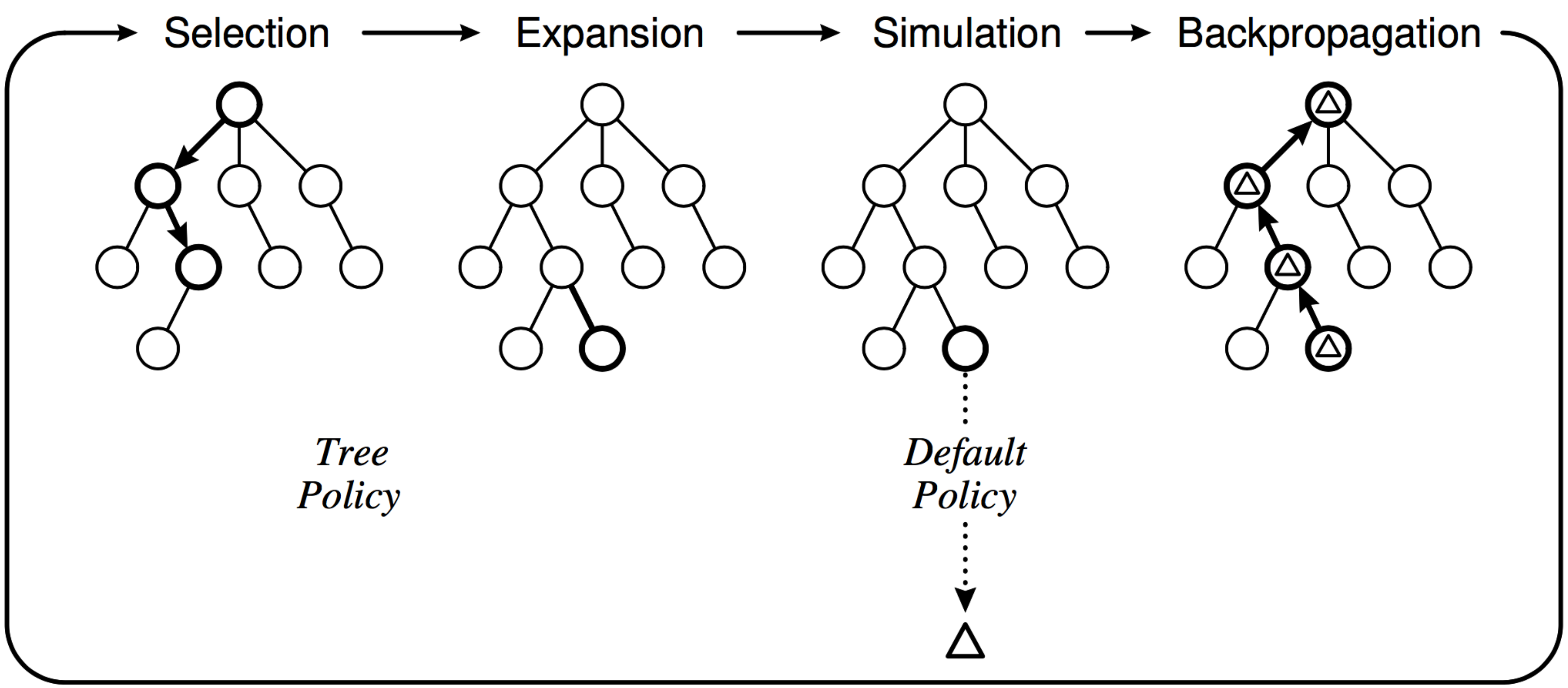
too costly

Informed Sampling

- Smarten the search to approximate the (near-) optimal product
  - using Monte Carlo Tree Search (MCTS)(Browne et al., 2012)

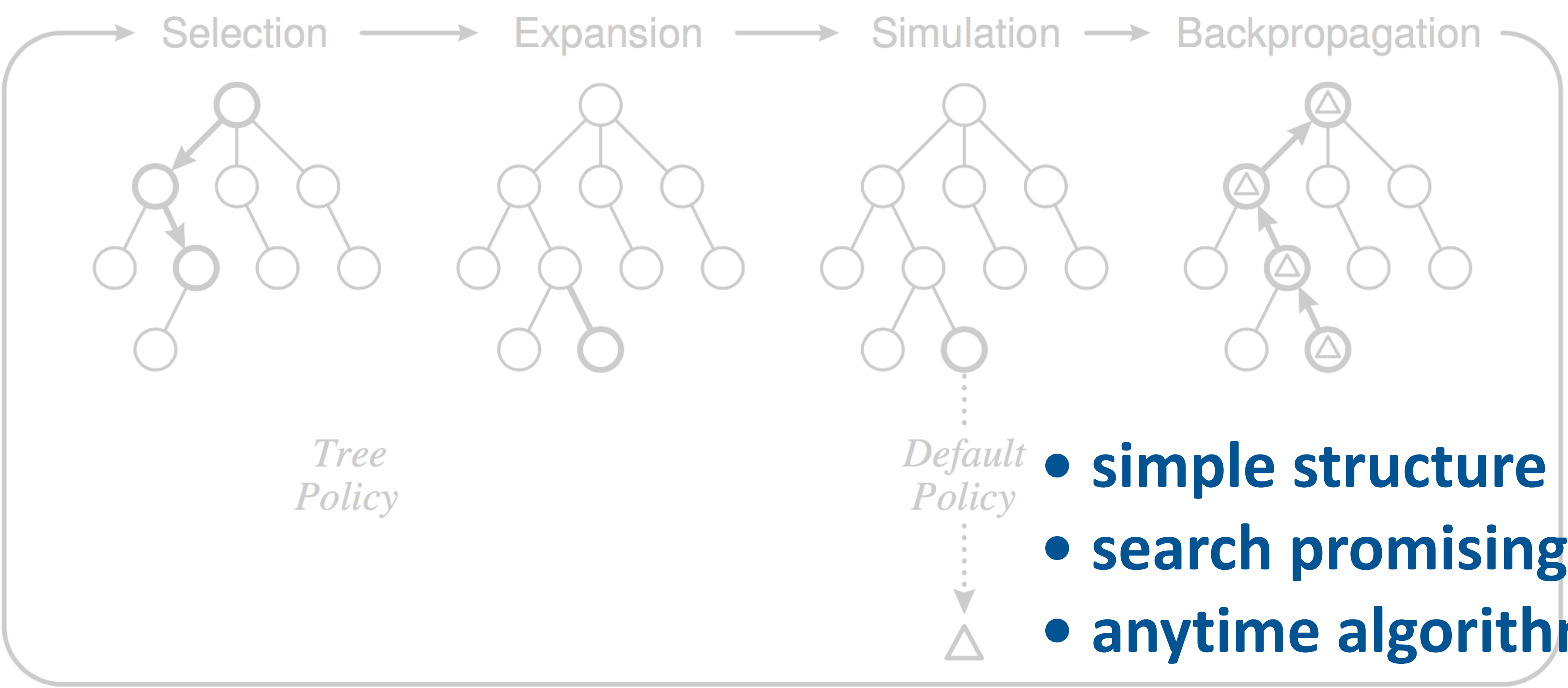
Informed Sampling

- Smarten the search to approximate the (near-) optimal product
  - using Monte Carlo Tree Search (MCTS)(Browne et al., 2012)



Informed Sampling

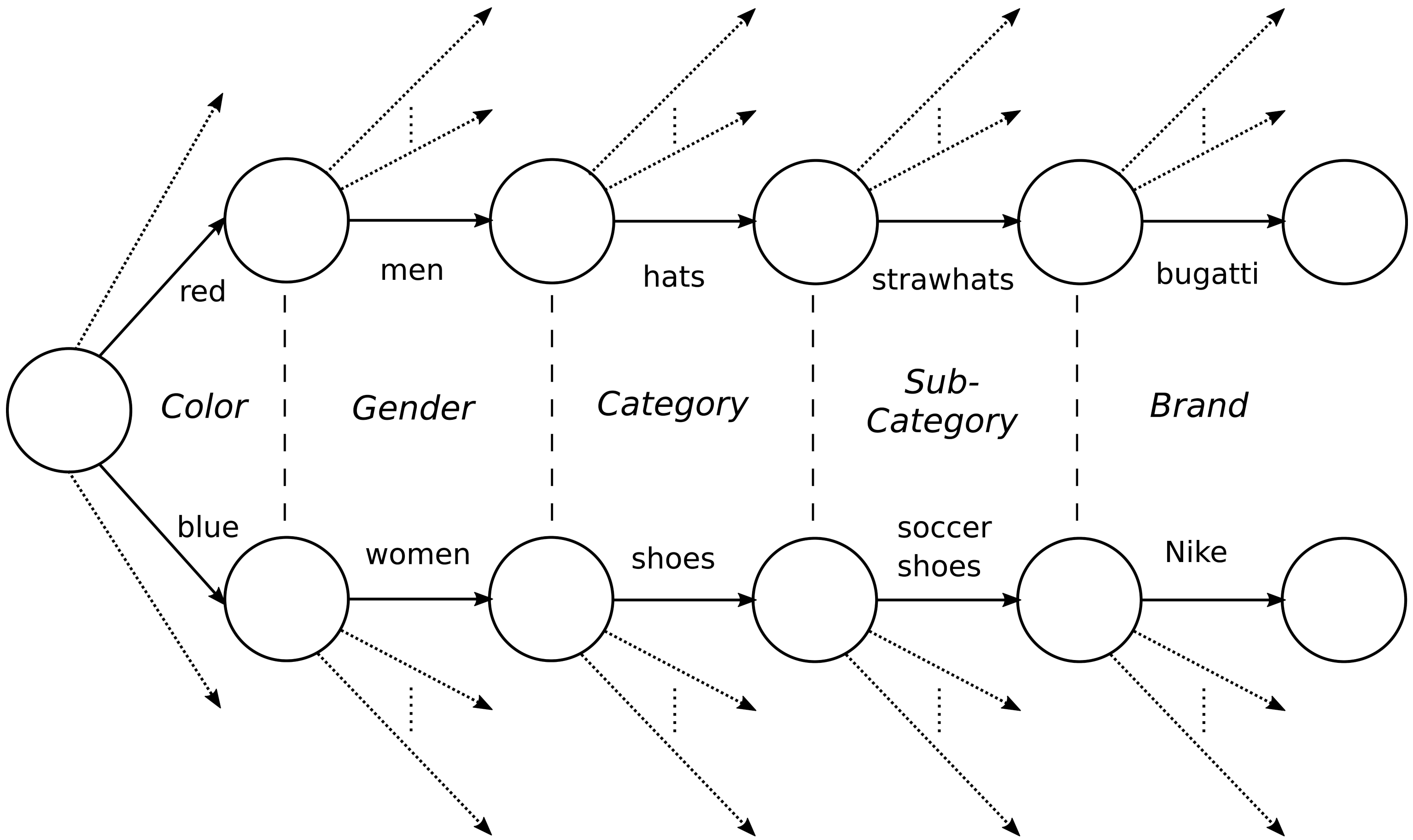
- Smarten the search to approximate the (near-) optimal product
  - using Monte Carlo Tree Search (MCTS)(Browne et al., 2012)



- **simple structure**
- **search promising branches**
- **anytime algorithm**
- **historically successful**

MCTS in Feature Space

- Leaf nodes specify a complete item with all categorical features



each layer selects an attribute

MCTS Modifications

- Utilize the preference information over features

MCTS Modifications

- Utilize the preference information over features
- Find the best product (leaf node), instead of the best action in the root node

MCTS Modifications

- Utilize the preference information over features
- Find the best product (leaf node), instead of the best action in the root node
- Keep a list of best *n* products found while searching

MCTS Modifications

- Utilize the preference information over features
- Find the best product (leaf node), instead of the best action in the root node
- Keep a list of best *n* products found while searching
- Avoid redundant evaluation of the same product
  - Remove leaf nodes that have been tested form the tree

Outline

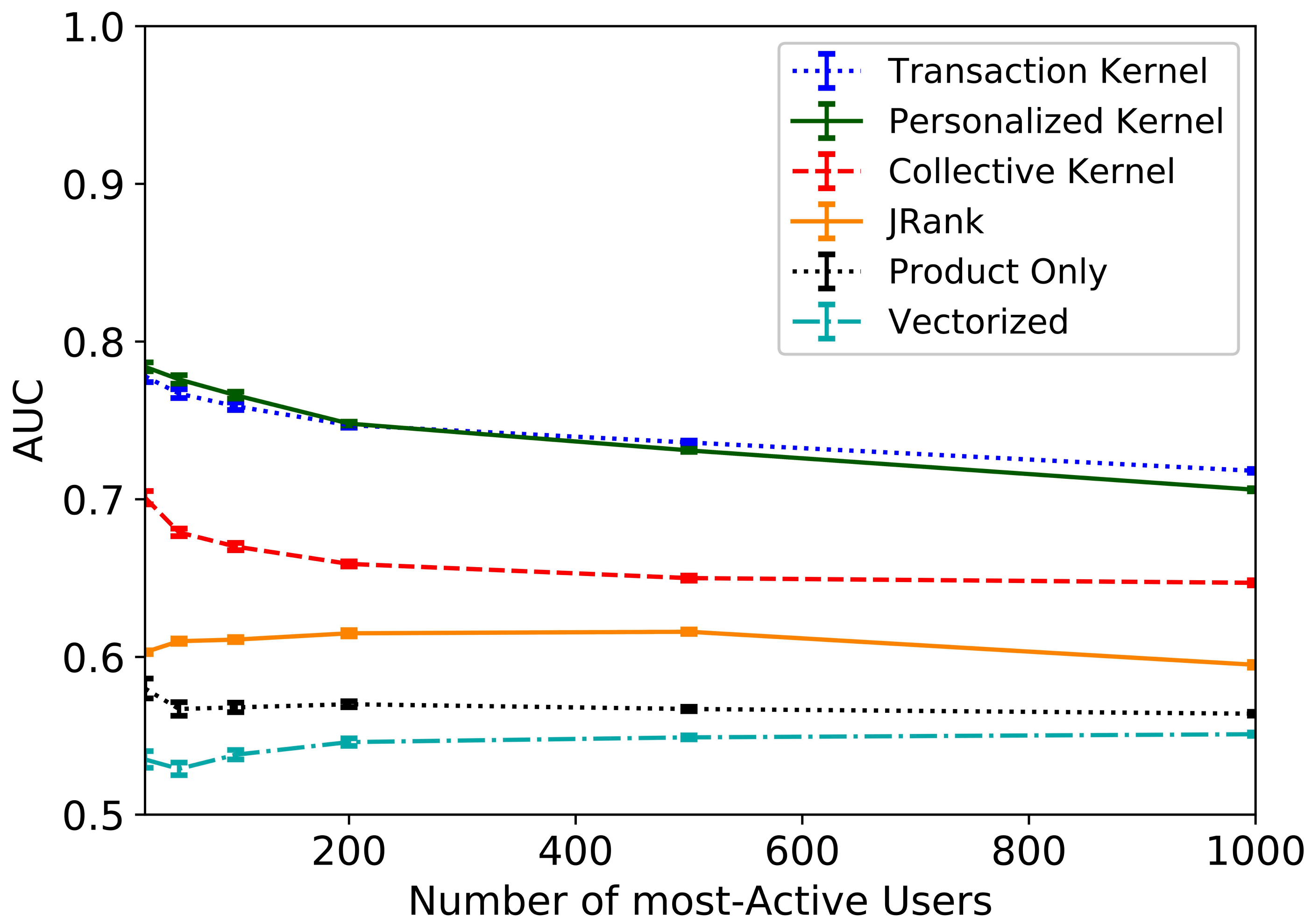
- Introduction and motivation
- Personalized transaction kernels
- Informed sampling strategies
- Empirical study
- Summary and Conclusions

Empirical Study

- Pairwise preference data from Zalando (www.zalando.com)
- About 680k users and 16k products
- More than 3.5 million total transactions
- **Attributes:** category, sub-category, color, gender, brand
- **Baseline:** JRank(Basilico, Hofmann, 2004)(preference model), random subset exhaustion and greedy stochastic search (search)
- **Evaluation measure:** AUC (preference model), utility (search)

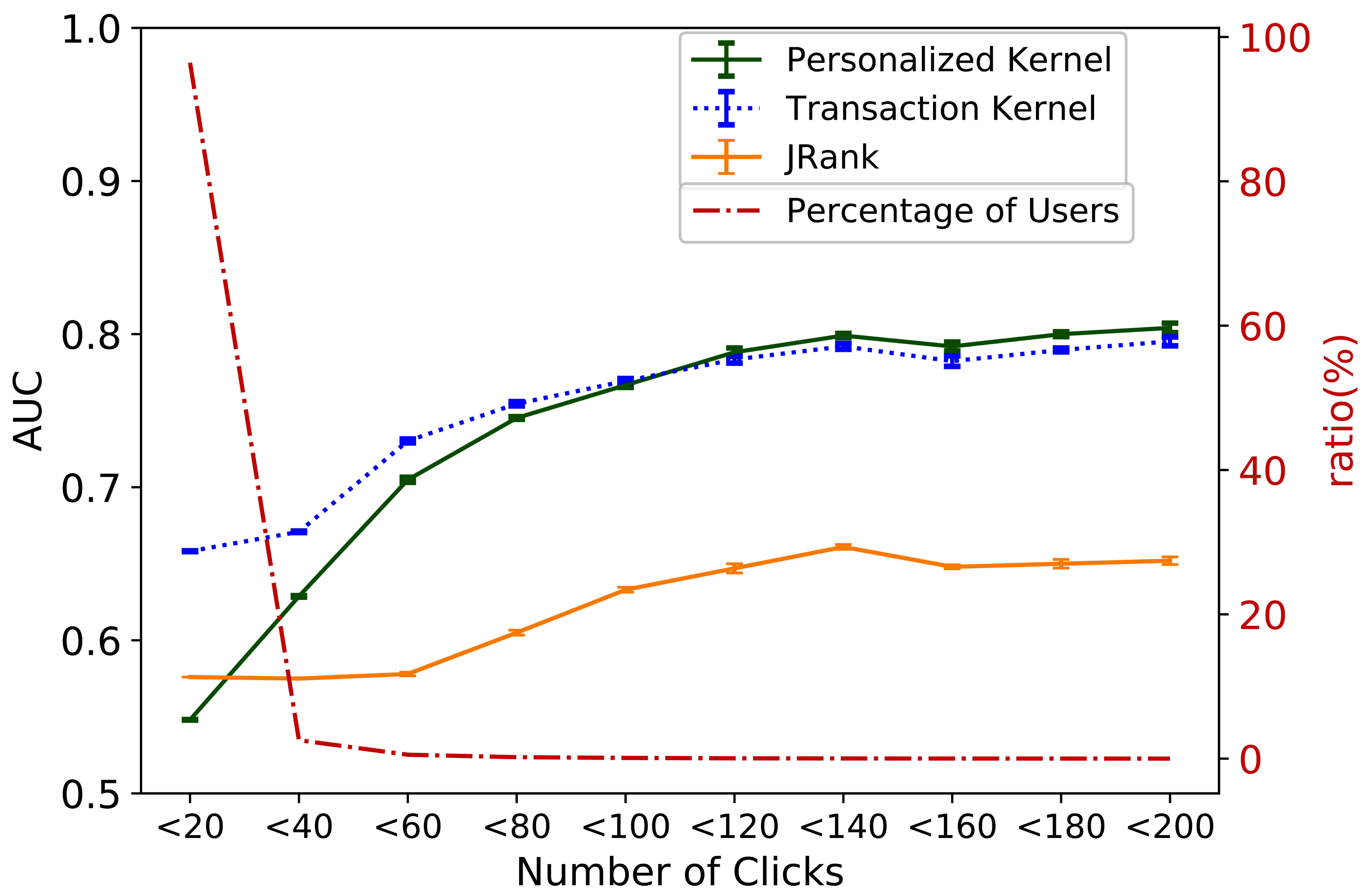
Results—Preference Model

Overall performance for 10-fold cross validation



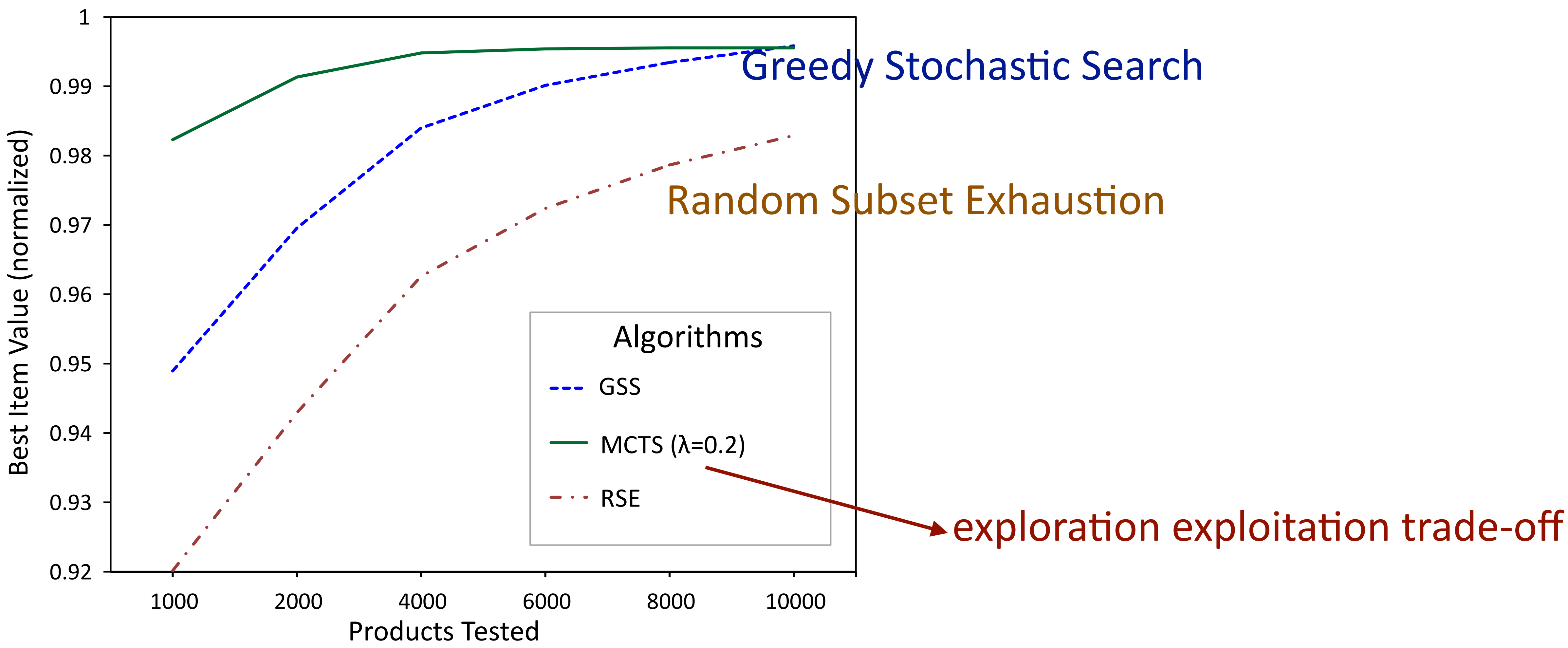
- highly accurate performance of personalized & combined model

Results—Preference Model



- more data per user: more robust personalization
  - but for most users, **collective kernel** has an important role

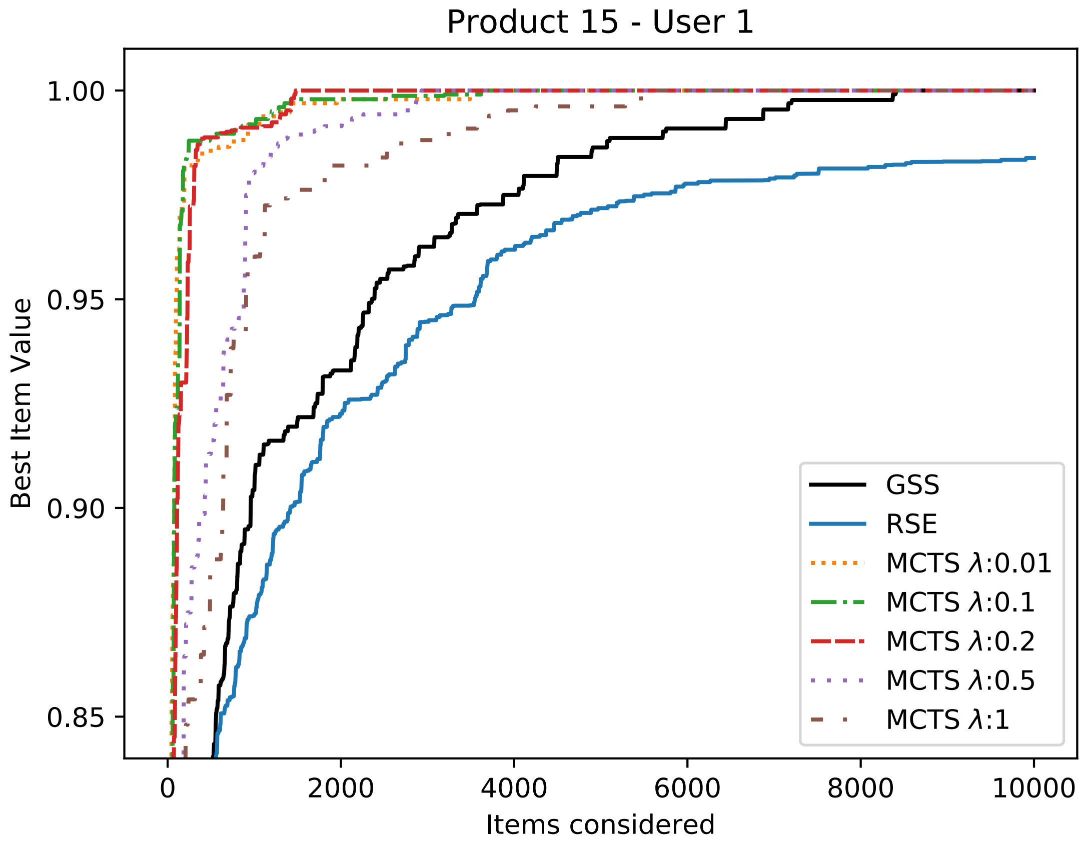
Results—Informed Sampling

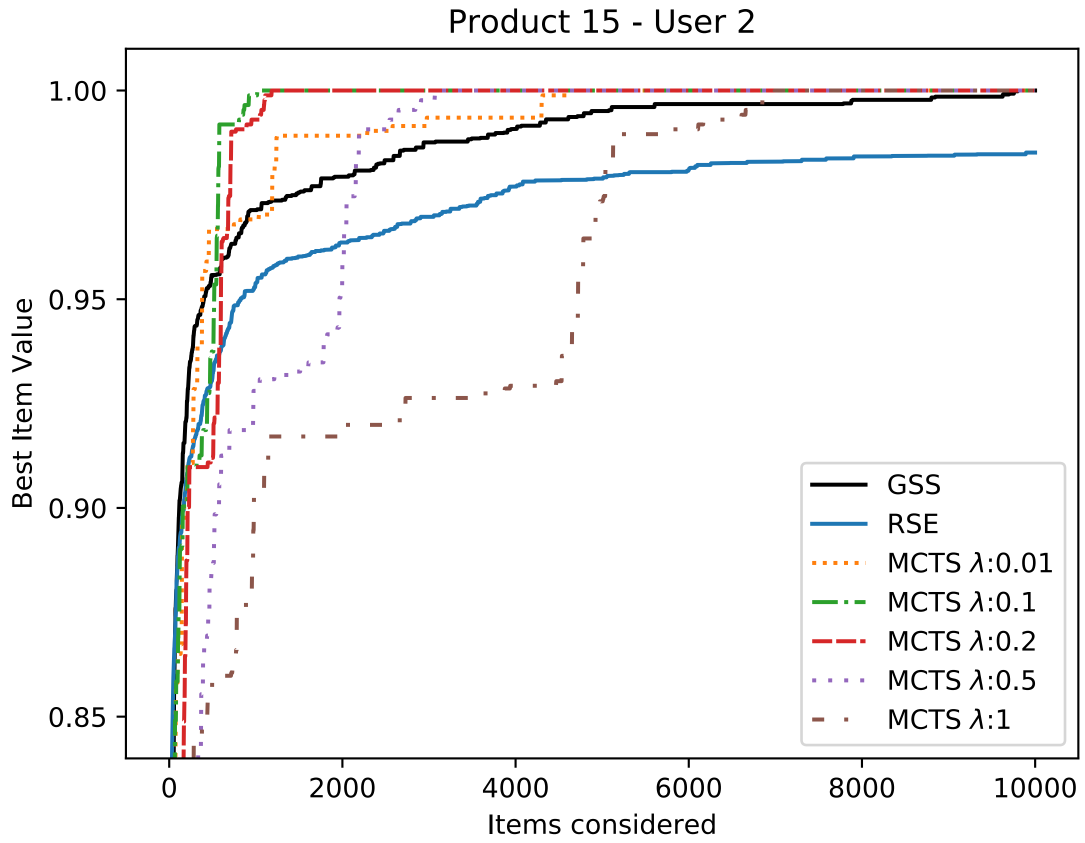


- MCTS outperforms the baselines in much smaller search spaces

Results—Informed Sampling

Example tests for a given product, four different users

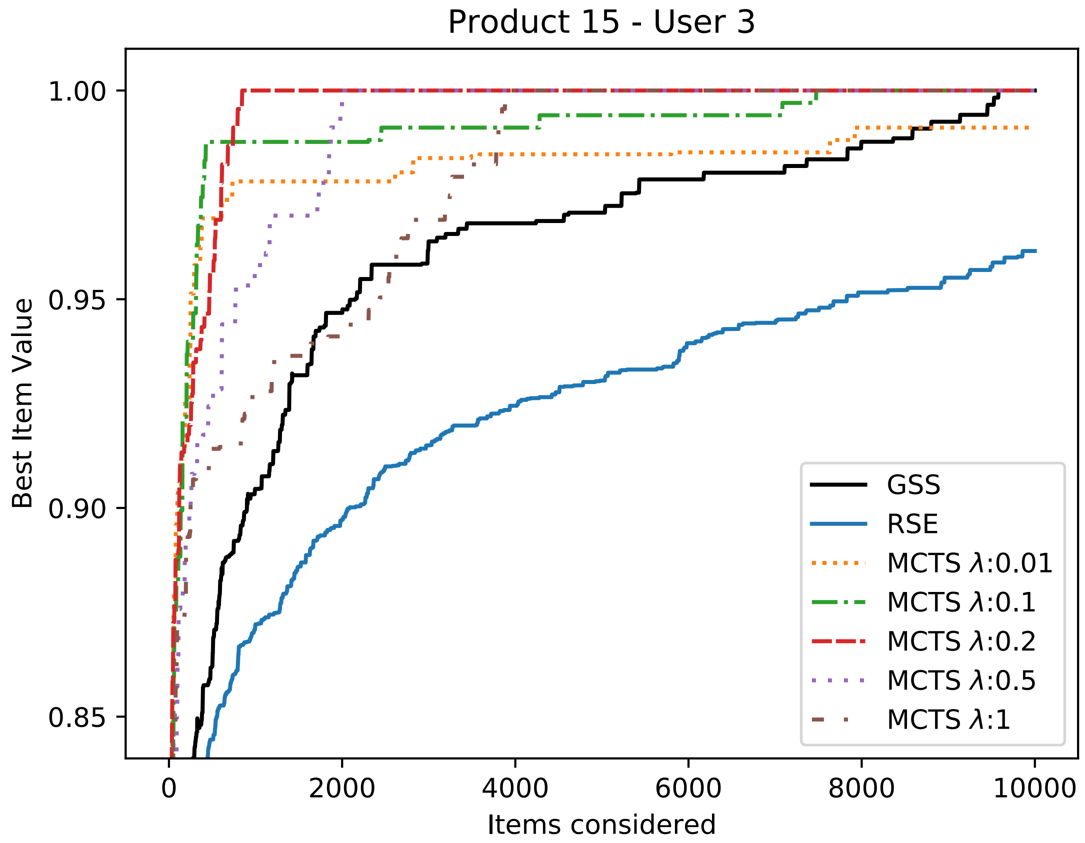


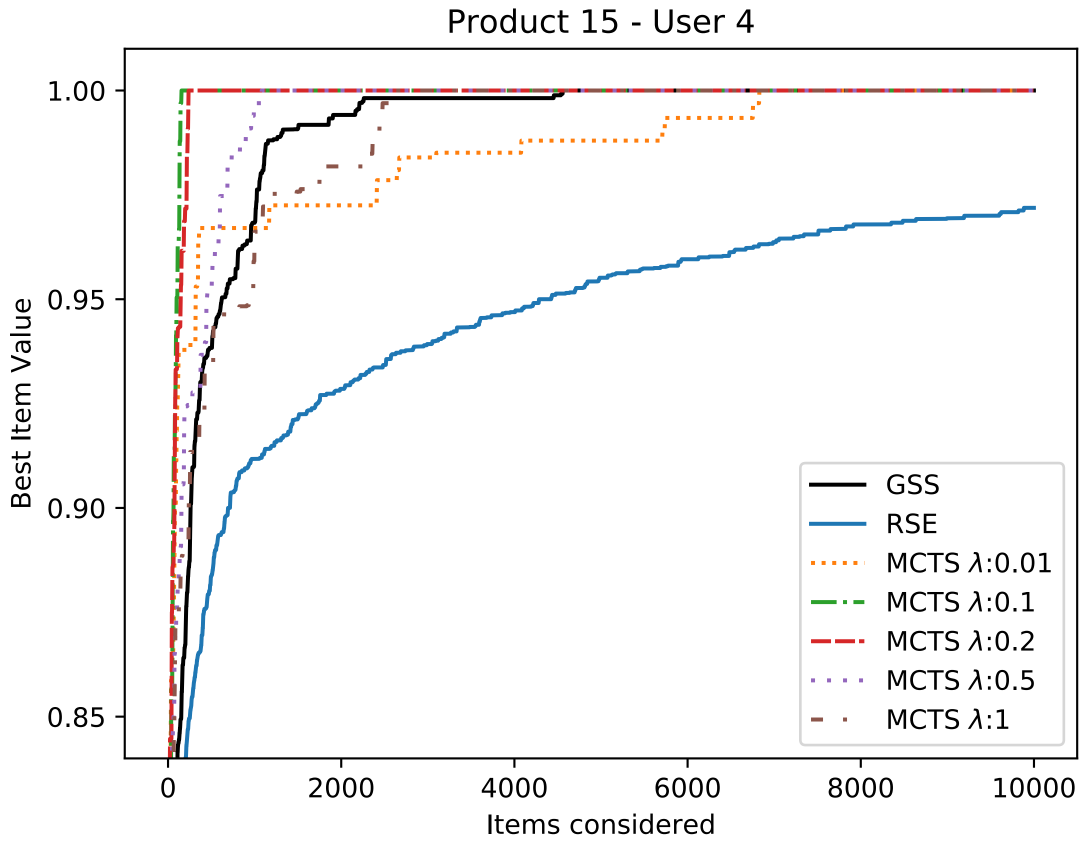


- the choice of λ is important in the performance

Results—Informed Sampling

User 4 is a new user without training data





- MCTS finds the optimal product with a lower number of samples

Outline

- Introduction and motivation
- Personalized transaction kernels
- Informed sampling strategies
- Empirical study
- Summary and Conclusions

Summary and Conclusions

Summary and Conclusions

- Preference-based learning to model personalized interests of users in contextual settings

Summary and Conclusions

- Preference-based learning to model personalized interests of users in contextual settings
- Transaction kernels combine **tensor products** with **hashing functions** to capture individual as well as collective user preferences

Summary and Conclusions

- Preference-based learning to model personalized interests of users in contextual settings
- Transaction kernels combine **tensor products** with **hashing functions** to capture individual as well as collective user preferences
- A variant of MCTS method efficiently retrieves near-optimal items for online recommendation

Summary and Conclusions

- Preference-based learning to model personalized interests of users in contextual settings
- Transaction kernels combine **tensor products** with **hashing functions** to capture individual as well as collective user preferences
- A variant of MCTS method efficiently retrieves near-optimal items for online recommendation
- The model shows effective and efficient performance in real-world scenarios

Thanks for Your Attention!

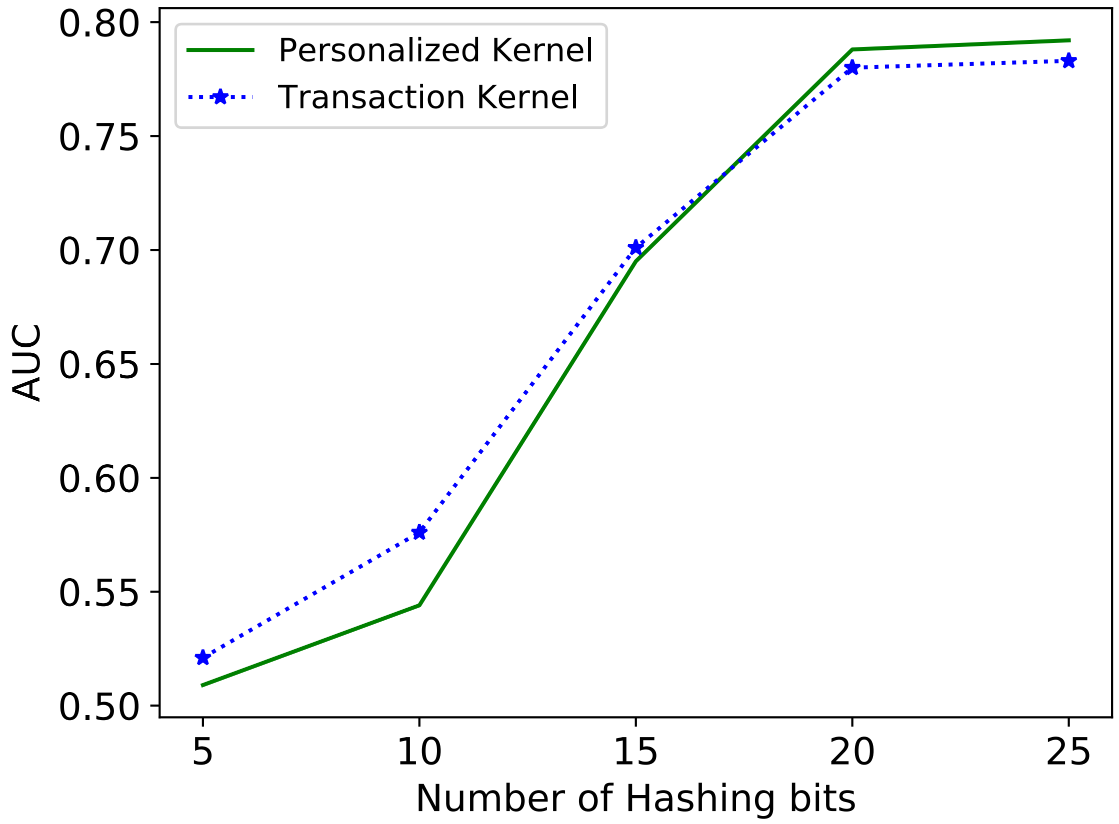
maryamtavakol.com
maryam.tavakol@tu-dortmund.de

- References

- P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. *Grouplens: an open architecture for collaborative filtering of netnews*. In Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work, pp. 175–186. ACM, 1994
- T. Joachims, *Optimizing search engines using clickthrough data*. In Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 133–142. ACM (2002)
- K. Weinberger, A. Dasgupta, J. Langford, A. Smola, J. Attenberg, *Feature hashing for large scale multitask learning*. In Proceedings of the 26th International Conference on Machine Learning (ICML-09). pp. 1113–1120. ACM (2009)
- C. B. Browne, E. Powley, D. Whitehouse, S. M. Lucas, P. I. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, S. Colton, *A survey of Monte Carlo tree search methods*. IEEE Transactions on Computational Intelligence and AI in Games 4(1), 1–43 (2012)
- J. Basilico, T. Hofmann, *Unifying collaborative and content-based filtering*. In Proceedings of the 21st International Conference on Machine Learning (ICML-04). p. 9. ACM (2004)

Results—Preference Model

The effect of the size of hashing function



- The more bits are used, the better the personalization model