Explainable Clustering and Cluster-based Collaborative Filtering

Yuntao Du and Jianxun Lian

Microsoft Research Asia

2023/1/10

Outline

- Explainable Clustering
  - Why clustering?
  - Why explainability is matter for clustering algorithms?
  - How to achieve explainable clustering?

- Cluster-based Collaborative Filtering
  - Revisiting: collaborative filtering as an explicit/implicit clustering process
  - Naïve clustering approach for collaborative filtering
  - Our work: Towards Explainable Collaborative Filtering with Taste Clusters Learning

Part1: Explainable Clustering

Clustering

- Why clustering?
  - One of the biggest topics in data science
  - Clustering is to identify patterns or discover structural properties in a data set by quantizing the unlabeled points
    - Discover coherent groups among a supermarket's customers
    - Find friends with similar habits / tastes
- A classical problem, but still plenty need to be done…
  - Classical clustering algorithms (*k*-means, DBSCAN, etc)
  - Co-clustering / constrained clustering / multi-view clustering
  - Deep clustering…

Explainable Clustering

- Explainability for clustering
  - Cluster process may be determined using <u>all the features </u>of the data or <u>embeddings</u>
    - Why they need to be in the same cluster?
  - Cluster results can be hard to explain
    - What’s the meaning of the cluster?
  - Quality is not the only objective in many fields like healthcare

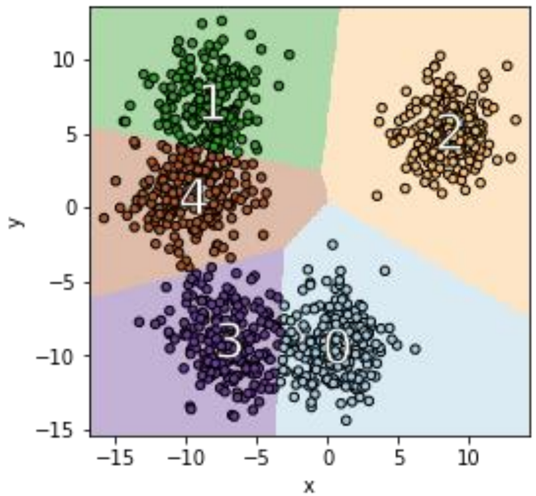
- Goal
  - Provide provable insight into what parts of the data the cluster algorithm used to make its prediction
  - Achieve good balance between quality and explainability

Tree-based explainable clustering

- k-means (reference clustering)
  - Cluster goal: minimize *k-means cost* (NP-hard)

$$cost(C) = \sum_{i=1}^{k} \sum_{x \in C^i} \|x - mean(C^i)\|^2$$

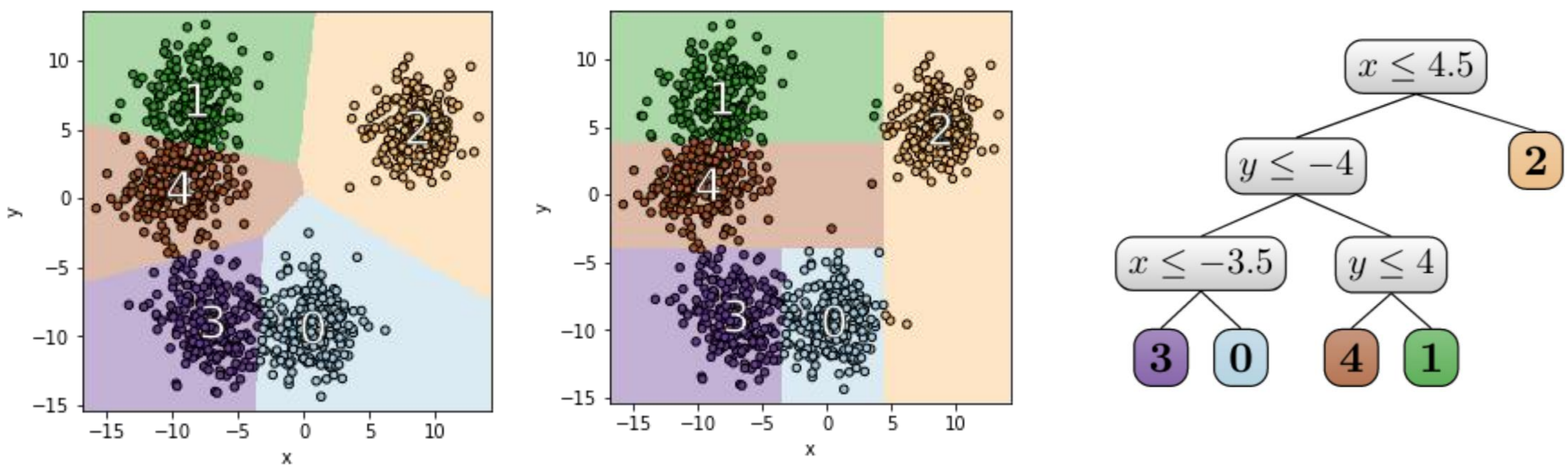
- Naive explanation
  - Directly using cluster centers as explanation
  - Depends on all data points and all the features in a complicated way



Sanjoy Dasgupta et al. "Explainable k-Means and k-Medians Clustering", ICML'20.

Tree-based explainable clustering

- Goal
  - Explainable by design (self-explainable)
  - Only look at some determined features to make clusters
    - Not depend on the cluster centers
  - At each step, split only one feature with threshold
    - Leaves correspond to clusters (same as decision tree)

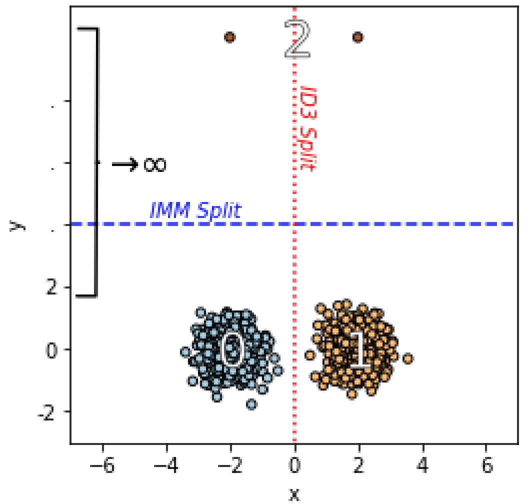


Sanjoy Dasgupta et al. "Explainable k-Means and k-Medians Clustering", ICML'20.

Tree-based explainable clustering

- General scheme
  - Find a clustering using some clustering algorithm
  - Label each example according to its cluster
  - Call a supervised algorithm that learns a decision tree
- ID3/C4.5 algorithm?
  - Split according to the information gain, no good

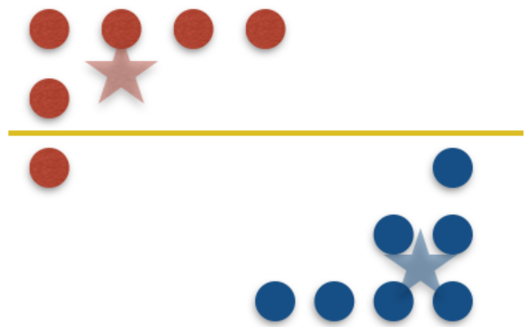
$$Gain(D, a) = Ent(D) - \sum_{t=1}^{T} \frac{|D^t|}{|D|} Ent(D^t)$$



Sanjoy Dasgupta et al. “Explainable k-Means and k-Medians Clustering”, ICML’20.

Tree-based explainable clustering

- Iterative Mistake Minimization (IMM)
  - Mistake
    - A point x is a mistake for node u if x and its center $c(x)$ reached and then separated by u's split

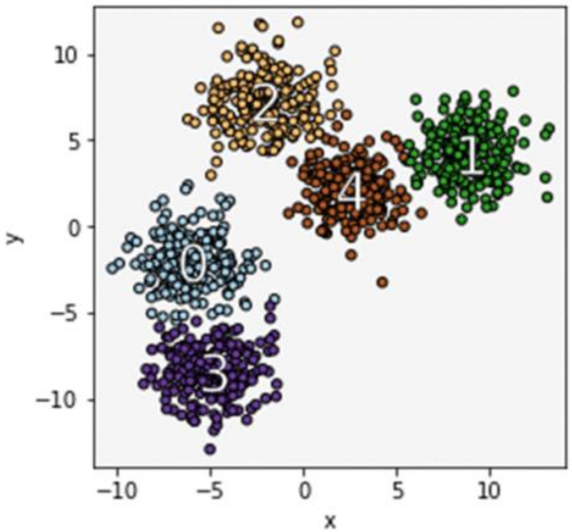


  - Each step we take the split (i.e., feature and threshold) that minimizes mistake

Sanjoy Dasgupta et al. “Explainable k-Means and k-Medians Clustering”, ICML’20.

Tree-based explainable clustering

- Iterative Mistake Minimization (IMM)
  - As long as there is more than one center, find the split with minimal number of mistakes



Mistakes

Current split: -

Total: -

Sanjoy Dasgupta et al. “Explainable k-Means and k-Medians Clustering”, ICML’20.

Tree-based explainable clustering

IMM Properties

- Running time
  - $O(kdn\log(n))$
  - For each of the $k-1$ inner nodes and each of the d features, we can find the split that minimizes the number of mistakes for this node and feature, in time $O(n\log(n))$
  - Comparable to standard k-means $O(tkdn)$
- Approximation factor

	k-medians		k-means	
	$k=2$	$k>2$	$k=2$	$k>2$
Lower	$2-\frac{1}{d}$	$\Omega(\log k)$	$3\left(1-\frac{1}{d}\right)^2$	$\Omega(\log k)$
Upper	2	$O(k)$	4	$O(k^2)$

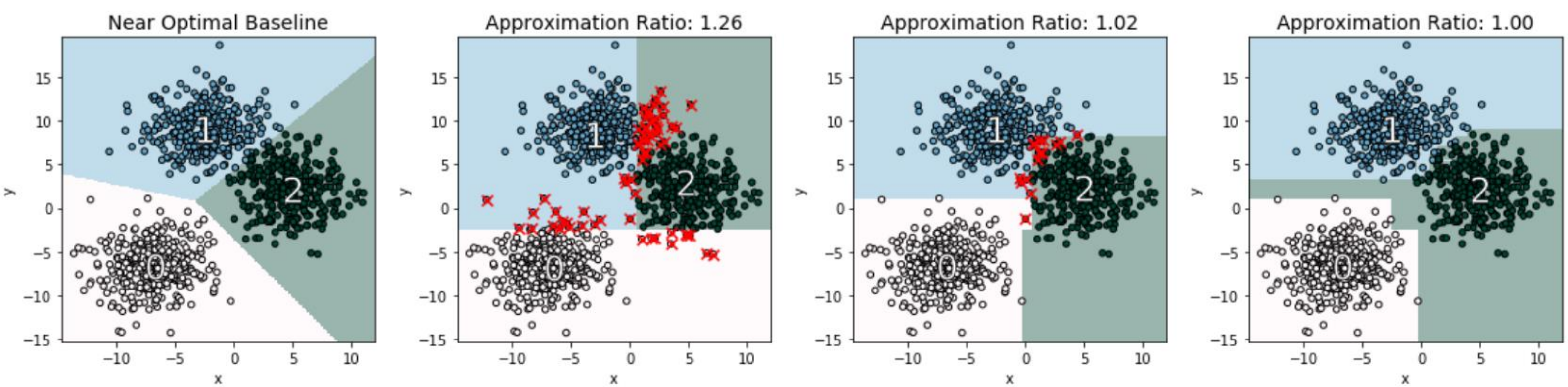
https://ucsdml.github.io/jekyll/update/2020/10/16/explain_2_means.html

Sanjoy Dasgupta et al. “Explainable k-Means and k-Medians Clustering”, ICML’20.

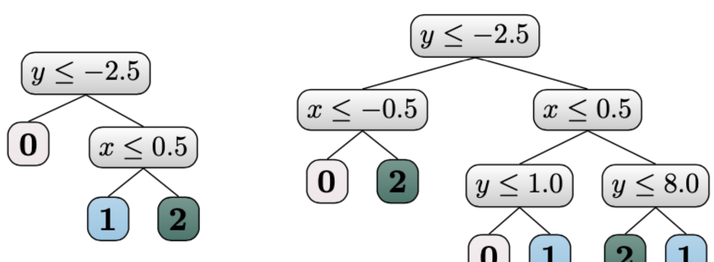
Tree-based explainable clustering

Improvements 1

- What if we can have k' leaves ($k' > k$)?
  - Flexible trade-off between explainability & accuracy



(a) k-means++ (reference) (b) k leaves (IMM) (c) $2k$ leaves (ExKMC) (d) n leaves



(e) IMM tree

(f) ExKMC tree

Nave Frost et al. “ExKMC: Expanding Explainable k-Means Clustering”, arXiv’20.

Tree-based explainable clustering

Improvements 2

- Can we make full use of the reference clustering?
  - Select cuts based on centers, not data points
  - Only need to scan data once, nearly zero computational overhead

Algorithm 1: Explainable k-medians algorithm.

Input: A collection of k centers $\mathcal{U} = \{\boldsymbol{\mu}^1, \boldsymbol{\mu}^2, \ldots, \boldsymbol{\mu}^k\} \subset \mathbb{R}^d$.
Output: A threshold tree with k leaves.
Leaves $\leftarrow \{\mathcal{U}\}$
while $|\text{Leaves}| < k$ **do**
  Sample (i, θ) uniformly at random from AllCuts.
  for *each* $B \in$ Leaves *that are split by* (i, θ) **do**
    Split B into B^- and B^+ and add them as left and right children of B.
    Update Leaves.
return the threshold tree defined by all cuts that separated some B.

$$\text{AllCuts} = \{(i, \theta) : i \in [d], \theta \in \mathrm{I}_i\}$$

Buddhima Gamlath et al. "Nearly-Tight and Oblivious Algorithms for Explainable Clustering", NeurIPS'21.

Tree-based explainable clustering

Improvements 2

- The constructed decision tree is almost good enough

Table 1: **Algorithms and lower bounds for explainable k-clustering in $\mathbb{R}^d$.** For a given objective function, how large a multiplicative factor do we have to lose, compared to an optimal unconstrained clustering, if we insist on an explainable clustering?

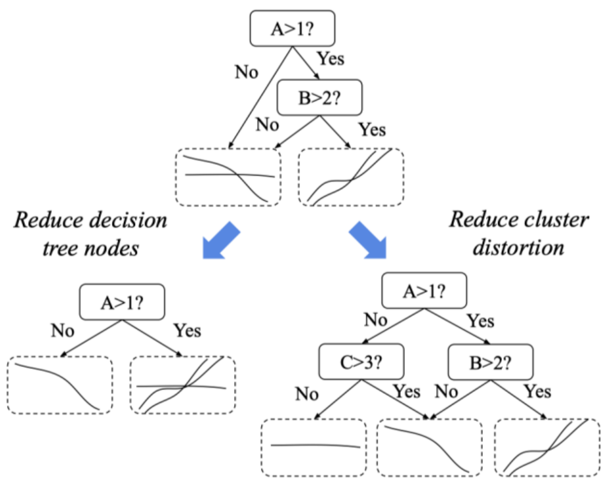
	k-medians	k-means	ℓ_p-norm	
Algorithms	$O(k)$	$O(k^2)$		Dasgupta et al. [6]
	$O(d \log k)$	$O(kd \log k)$		Laber and Murtinho [10]
	$O(\log^2 k)$	$O(k \log^2 k)$	$O(k^{p-1} \log^2 k)$	**This paper**
	$O(\log k \log\log k)$	$O(k \log k \log\log k)$		Makarychev and Shan [12]
	$O(\log k \log\log k)$	$O(k \log k)$		Esfandiari et al. [7]
	$O(d \log^2 d)$			Esfandiari et al. [7]
		$O(k^{1-2/d} \operatorname{polylog} k)$		Charikar and Hu [5]
Lower bounds	$\Omega(\log k)$	$\Omega(\log k)$		Dasgupta et al. [6]
		$\Omega(k)$	$\Omega(k^{p-1})$	**This paper**
		$\Omega(k/\log k)$		Makarychev and Shan [12]
	$\Omega(\min(d, \log k))$	$\Omega(k)$		Esfandiari et al. [7]
		$\Omega(k^{1-2/d} / \operatorname{polylog} k)$		Charikar and Hu [5]

Buddhima Gamlath et al. “Nearly-Tight and Oblivious Algorithms for Explainable Clustering”, NeurIPS’21.

Tree-based explainable clustering

Improvements 3

- Can perform clustering and decision tree training holistically?
  - Optimize the decision tree's size (for explainability) and the distortion (for accuracy) together
  - Assume two groups of features: accuracy features and explainability features
  - Clustering with accuracy features, analyze with explainability features



Hyunseung Hwang et al. "XClusters: Explainability-first Clustering", AAAI'23.

Tree-based explainable clustering

Problem setting

- Define similarity functions as the combination of two measures

$$\frac{(1-\alpha)\times \text{a-distance}}{\max\{\text{All a-distances}\}} + \frac{\alpha \times \text{e-distance}}{\max\{\text{All e-distances}\}}$$

- Can use any distance function (DTW) and clustering method (k-medoids)
- Agnostic to the decision tree training algorithm

Goal

$$\min_{k,\alpha} \; D(k,\alpha) + \lambda N(k,\alpha)$$

- D: cluster distortion where a lower value is better as we would like the clusters to be coherent
- N: number of decision tree nodes

Hyunseung Hwang et al. “XClusters: Explainability-first Clustering”, AAAI’23.

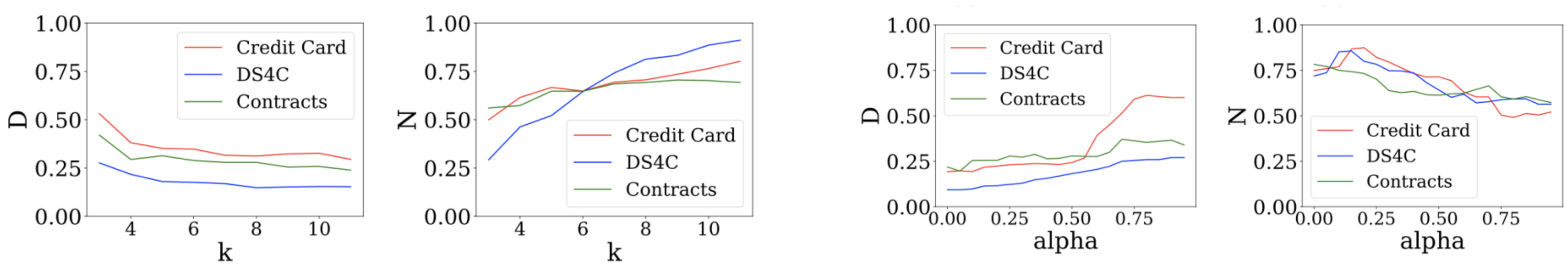
Tree-based explainable clustering

Monotonicity properties

- As k increases, D is decreasing while N is increasing
- As α increases, D is increasing while N is decreasing
- Given $[k_1, k_2]$ and $[\alpha_1, \alpha_2]$, we have

$$\begin{aligned} & D(k,\alpha) + \lambda N(k,\alpha) \\ \geq\ & D(k_2,\alpha) + \lambda N(k_1,\alpha) \\ \geq\ & D(k_2,\alpha_1) + \lambda N(k_1,\alpha_2). \end{aligned}$$

- Get the upper bound and lower bound for $D(k,\alpha) + \lambda N(k,\alpha)$
- Search for the desired parameters



(a) k versus D (b) k versus N (c) α versus D (d) α versus N

Hyunseung Hwang et al. "XClusters: Explainability-first Clustering", AAAI'23.

Tree-based explainable clustering

Search for the k and α parameters

Algorithm 1: XClusters algorithm

Input: training data S, maximum k value $k_{\max}$
Parameters: k, α
Output: clusters and decision tree

$B \leftarrow [(1,0),(k_{\max},1)]$
Compute upper and lower bounds of B
$B^* \leftarrow B$
$Q.push(B)$
while $\neg Q.empty()$ **do**
  $B \leftarrow Q.pop()$ // Block with lowest lower bound
  if B's normalized k width is longer than the normalized α width **then**
    $\{B_1, B_2\} \leftarrow$ Split B by k into two blocks
  else
    $\{B_1, B_2\} \leftarrow$ Split B by α into two blocks
  Compute upper and lower bounds of B_1 and B_2
  $Q.push(\{B_1, B_2\})$
  if $\min_{B \in Q} B.upper() < B^*.upper()$ **then**
    $B^* \leftarrow \arg\min_{B \in Q} B.upper()$
  $Q \leftarrow Q \setminus \{B' \in Q | B'.lower() + \epsilon_b \geq B^*.upper()\}$
return Clusters and decision tree of $B^*.upper()$

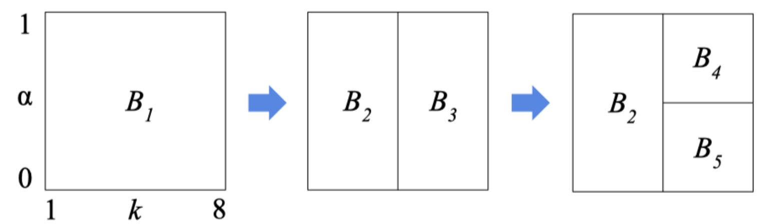


Figure 2: The XClusters algorithm iteratively splits blocks while pruning blocks that are not worth exploring based on their lower and upper bounds.

Hyunseung Hwang et al. "XClusters: Explainability-first Clustering", AAAI'23.

Tree-based explainable clustering

Experiments

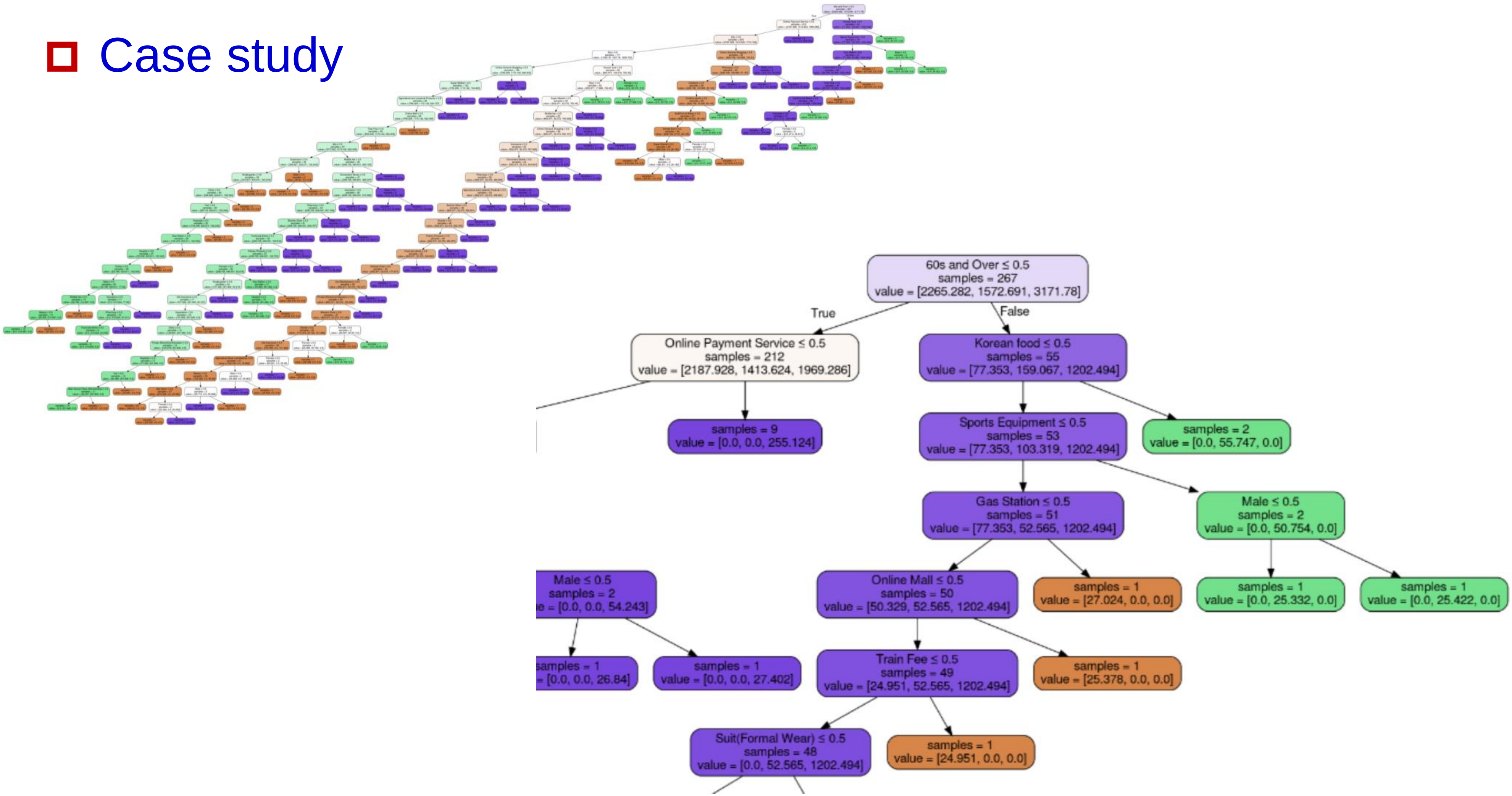
- Use three time series dataset: Credit, COVID-19, contracts
- Accuracy features: time-series trends features
- Explainability features: demographics information
- Achieve good balance between accuracy and explainability, as well as efficiency

Dataset	Method	$D + \lambda N$	D	N	Runtime (sec)
Credit Card	2-Step	$0.981_{\pm 0.000}$	$0.206_{\pm 0.000}$	$0.775_{\pm 0.000}$	$0.011_{\pm 0.001}$
	GS	$0.808_{\pm 0.000}$	$0.282_{\pm 0.000}$	$0.526_{\pm 0.000}$	$1.413_{\pm 0.023}$
	BO	$0.851_{\pm 0.028}$	$0.238_{\pm 0.030}$	$0.613_{\pm 0.045}$	$1.416_{\pm 0.158}$
	XClusters	$0.815_{\pm 0.000}$	$0.270_{\pm 0.000}$	$0.544_{\pm 0.000}$	$0.321_{\pm 0.004}$
DS4C	2-Step	$0.702_{\pm 0.000}$	$0.103_{\pm 0.000}$	$0.599_{\pm 0.000}$	$0.008_{\pm 0.001}$
	GS	$0.406_{\pm 0.000}$	$0.127_{\pm 0.000}$	$0.279_{\pm 0.000}$	$1.345_{\pm 0.012}$
	BO	$0.415_{\pm 0.026}$	$0.126_{\pm 0.005}$	$0.289_{\pm 0.031}$	$1.546_{\pm 0.225}$
	XClusters	$0.466_{\pm 0.000}$	$0.128_{\pm 0.000}$	$0.338_{\pm 0.000}$	$0.308_{\pm 0.004}$
Contracts	2-Step	$1.018_{\pm 0.000}$	$0.216_{\pm 0.000}$	$0.802_{\pm 0.000}$	$0.011_{\pm 0.000}$
	GS	$0.778_{\pm 0.000}$	$0.228_{\pm 0.000}$	$0.550_{\pm 0.000}$	$1.974_{\pm 0.000}$
	BO	$0.827_{\pm 0.032}$	$0.301_{\pm 0.057}$	$0.526_{\pm 0.046}$	$1.550_{\pm 0.240}$
	XClusters	$0.619_{\pm 0.000}$	$0.442_{\pm 0.000}$	$0.177_{\pm 0.000}$	$0.349_{\pm 0.004}$

Hyunseung Hwang et al. “XClusters: Explainability-first Clustering”, AAAI’23.

Tree-based explainable clustering

- Case study



Hyunseung Hwang et al. “XClusters: Explainability-first Clustering”, AAAI’22.

Explainable clustering

- Future direction
  - Efficiency
    - Can we explore parallelizations for threshold tree construction?
  - Generalization
    - Can we allow each node to be a hyperplane in a chosen number of dimensions instead of only splitting along one feature?
  - Evaluation
    - How to evaluate the quality of explainability?
  - Other explainable approaches
    - Can we go beyond the decision tree style for clustering explanation?

Part2: Cluster-based Collaborative Filtering

CF as a clustering process

- Rationale behind collaborative filtering
  - Finding like-minded users for a (group of) target user(s) to share preferences
  - Much like clustering!

- Both memory-based CF (kNN, itemCF) and model-based CF (MF, NCF) perform clustering explicitly or implicitly
  - Directly cluster similar users/items into groups, then perform kNN for recommendation
  - Learn to group similar users/items implicitly, then select most similar items for recommendation

CF as a clustering process

- Naïve approach
  - Clustering users with any cluster algorithm
  - Data smoothing with cluster-specific rating (optional)

$$\hat{R}_u(t) = \overline{R_u} + \Delta R_{C_u}(t) \qquad \Delta R_{C_u}(t) = \sum_{u' \in C_u(t)} (R_{u'}(t) - \overline{R_{u'}}) / |C_u(t)|$$

  - Neighbor pre-selection with clusters
    - Select most similar group for active user

$$sim_{u_a,C} = \frac{\sum_{t \in T(u_a) \wedge T(C)} \Delta R_C(t) \cdot (R_{u_a}(t) - \overline{R_{u_a}})}{\sqrt{\sum_{t \in T(u_a) \wedge T(C)} (\Delta R_C(t))^2} \sqrt{\sum_{t \in T(u_a) \wedge T(C)} (R_{u_a}(t) - \overline{R_{u_a}})^2}}.$$

  - Neighbor selection
    - Select most similar users from certain group
  - Prediction

$$R_{u_a}(t) = \overline{R_{u_a}} + \frac{\sum_{i=1}^{K} w_{ut} \cdot sim_{u_a,u} \cdot (R_u(t) - \bar{R}_u)}{\sum_{i=1}^{K} w_{ut} \cdot sim_{u_a,u}}$$

- Is it necessary to explicitly perform clusters NOW?
  - Yes!

Gui-Rong Xue et al. “Scalable Collaborative Filtering Using Cluster-based Smoothing ”, SIGIR’05.

CF as a clustering process

- A toy movie recommendation scenario

	Andre	Star Wars	Batman	Rambo	Hiver	Whispers
Lyle	y	y				
Ellen	y	y			y	
Fred		y	y			
Dean		y	y	y		
Jason					y	y

CF as a clustering process

- Rearrange the table…

	Batman	Rambo	Andre	Hiver	Whispers	Star Wars
Lyle			y			y
Ellen			y	y		y
Jason				y	y	
Fred	y					y
Dean	y	y				y

CF as a clustering process

- Rearrange the table…

		action		foreign			classical
		Batman	Rambo	Andre	Hiver	Whispers	Star Wars
intellectual	Lyle			y			y
	Ellen			y	y		y
	Jason				y	y	
fun	Fred	y					y
	Dean	y	y				y

CF as a clustering process

- Rearrange the table…

		action		foreign			classical
		Batman	Rambo	Andre	Hiver	Whispers	Star Wars
intellectual	Lyle			y			y
	Ellen			y	y		y
	Jason				y	y	
fun	Fred	y					y
	Dean	y	y				y

	action	foreign	classic
intellectual	0/6	5/9	2/3
fun	3/4	0/6	2/2

- How to achieve this?

Towards Explainable Collaborative Filtering with Taste Clusters Learning

Yuntao Du[1], Jianxun Lian [2], Jing Yao [2], Xiting Wang [2], Mingqi Wu[3], Lu Chen[1], Yunjun Gao[1] and Xing Xie [2]

[1]*College of Computer Science, Zhejiang University, China*

[2] *Microsoft Research Asia, China*

[3]*Microsoft Gaming, USA*

The WebCof 2023, Under Review

Motivations

Explainable Recommendation

- A growing need to ensure that the users understand and trust the system
- Explanations serve as a **bridge** between recommender systems and users/developers
  - Increase user trust
  - Help users make better decisions (satisfactions)
  - Persuade users to try or buy an item (persuasiveness)
  - Assisting developers in model debugging and abnormal case studies

Explanations: why the items are recommended

Desirable Properties for Explainable CF

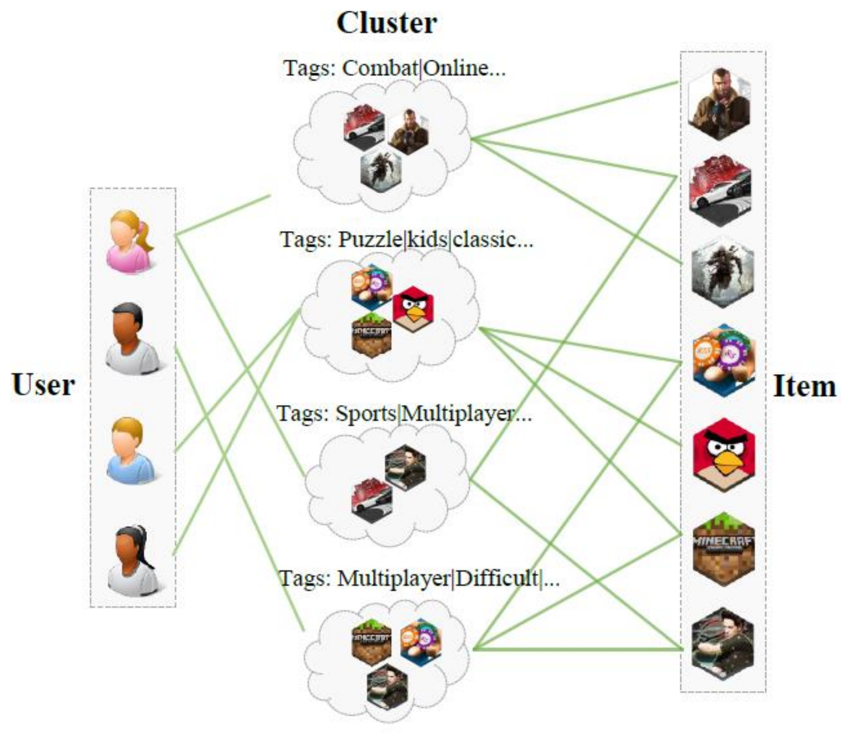
- Flexibility
  - The dimension of latent embeddings and the number of interpretable features/topics do not necessarily match ([1] fails on this)
- Coherence
  - A model's interpretable modules and predictive modules should be aligned during predictive decision making rather than being decoupled as independent modules ([2] fails on this)
- Self-explainable
  - A model can provide interpretable clues that truly reveal the model's running logic, instead of learning a post-hoc model for explanation ([3] fails on this)

[1] Julian McAuley et al. "Hidden factors and hidden topics: understanding rating dimensions with review text ", RecSys'13.
[2] Yongfeng Zhang et al. "Explicit factor models for explainable recommendation based on phrase-level sentiment analysis", SIGIR'14.
[3] Deng Pan et al. "Explainable Recommendation via Interpretable Feature Mapping and Evaluation of Explainability", IJCAI 20.

Our Method

Explainable Collaborative Filtering (ECF)

- The first framework that satisfies all three properties
- Core idea: mining various **taste clusters**, and map users/items to corresponding clusters
- **Taste clusters:** A group of items which are not only similar in users' latent interest space, but also explicitly share some common tags



Recommendation process of ECF

Item recommendation

- Prediction score of user u and item i can be calculated by multiplying their affiliations with taste clusters

$$\hat{y}_{ui} = \text{sparse\_dot}(\mathbf{a}_u, \mathbf{x}_i),$$

Personalized explanation

- For each prediction $\hat{y}_{ui}$, ECF is able to generate explanation by measuring the coherence between users' and items' taste cluster affiliations:

$$C_{ui} = S(\mathbf{a}_u) \cap S(\mathbf{x}_i),$$

- And importance score w_{ui}^c is introduced to quantify the contribution of each taste cluster in C_{ui}:

$$w_{ui}^c = a_{uc} \times x_{ic}.$$

Learning Sparse Affiliation

- Directly learning the affiliation matrix from data is hard
  - Due to its sparsity nature for readability

- Initialize the users/items and taste clusters with embedding

$$\tilde{x}_{ic} = \cos(\mathbf{v}_i, \mathbf{h}_c),$$

$$m_{ic} = \begin{cases} 1 & \text{if } c \in \operatorname{argTopm}(\tilde{\mathbf{x}}_i) \\ 0 & \text{otherwise} \end{cases}$$

$$\mathbf{x}_i = \sigma(\tilde{\mathbf{x}}_i) \odot \mathbf{m}_i,$$

- Learn it with reparameterized trick

$$m_{ic} \approx \tilde{m}_{ic} = \frac{\exp(\cos(\mathbf{v}_i, \mathbf{h}_c)/T)}{\sum_c \exp(\cos(\mathbf{v}_i, \mathbf{h}_c)/T)},$$

$$\hat{m}_{ic} = \tilde{m}_{ic} + \text{detach\_gradient}(m_{ic} - \tilde{m}_{ic}),$$

Optimization of ECF

Reconstruction Loss

- Using user/item-cluster affiliations for prediction:

$$\mathcal{L}_{\text{CS}} = \sum_{(u,i,j)\in\mathcal{O}} -\ln\sigma(\hat{y}_{ui} - \hat{y}_{uj}), \qquad \hat{y}_{ui} = \text{sparse\_dot}(\mathbf{a}_u, \mathbf{x}_i),$$

Tag Similarity Loss

- The items in the same taste clusters should share the similar tags
- Using TF-IDF style to select informative tags for taste clusters:

$$d_{ct} = \tilde{d}_{ct} \times \log(\frac{N}{f_t + \epsilon}), \quad \beta_{ct} = \frac{\exp(d_{ct}/\tau)}{\sum_{c_j\in\mathcal{T}} \exp(d_{ct}/\tau)},$$

- Maximizing the likelihood of the probabilities of Top-P tags so that the taste clusters can be easily interpreted by those tags:

$$\mathcal{L}_{\text{TS}} = \sum_{c\in C} \sum_{t\in \text{argTopP}(\beta_c)} -\log \beta_{ct},$$

Independence Loss

- Taste clusters should be different to present different user interest space:

$$\mathcal{L}_{\text{IND}} = \sum_{c\in C} -\log \frac{\exp(s(\mathbf{h}_c, \mathbf{h}_c))}{\sum_{c'\in C} \exp(s(\mathbf{h}_c, \mathbf{h}_{c'}))},$$

Optimization of ECF

- Learning taste cluster from three aspects
  - No need to tune the weight for each loss

$$\mathcal{L}_{\mathrm{TC}} = \mathcal{L}_{\mathrm{CS}} + \mathcal{L}_{\mathrm{TS}} + \mathcal{L}_{\mathrm{IND}}.$$

- ECF loss
  - Directly learning the taste cluster is hard to converge since the supervised signals are sparse
  - Adding auxiliary supervised signals from user-item predictions

$$\mathcal{L}_{\mathrm{CF}} = \sum_{(u,i,j)\in O} -\ln \sigma(\mathbf{e}_u^\top \mathbf{v}_i - \mathbf{e}_u^\top \mathbf{v}_j),$$

  - Embeddings can be learned from any embedding-based models (MF for simplicity)
  - Learn ECF with guidance from auxiliary collaborative signals

$$\mathcal{L}_{\mathrm{ECF}} = \mathcal{L}_{\mathrm{TC}} + \lambda \mathcal{L}_{\mathrm{CF}},$$

Forest Mechanism

Observation

- Sparse affiliations between user/item and clusters would inevitably harm recommendation accuracy
- We do not know how many clusters needed to model users' hidden interest space properly

Forest mechanism for ECF

- We randomly select $|C|$ items and use different random seeds for model training
- Train F different instances to form the final ECF model, and the final prediction is based on the summation of all M models
- Boost the performance and provide a comprehensive explanation for predictions

Metrics for Explainability

- In-cluster item coverage
  - The proportion of items in the taste cluster that the selected tags can cover

$$\text{Cov.} = \frac{1}{Z} \sum\nolimits_{c \in C} \sum\nolimits_{i \in c} \frac{\mathbb{1}(\mathcal{T}_i \cap \mathcal{T}_c)}{|c|},$$

- Tag utilization
  - How many unique tags are used for interpreting taste clusters

$$\text{Util.} = \frac{1}{|\mathcal{T}|} \bigcup_{c \in C} \mathcal{T}_c,$$

- Silhouette
  - Similarity difference between intra-cluster items and inter-cluster items

$$\text{Sil.} = \frac{1}{|\mathcal{I}|} \sum\nolimits_{i \in \mathcal{I}} \frac{b(i) - a(i)}{\max\{a(i), b(i)\}},$$

- Informativeness
  - Distinctiveness of selected tags to represent the items in the taste cluster

$$\text{Info.} = \frac{1}{|C|} \sum\nolimits_{c_i \in C} \frac{|R(\mathcal{T}_c) \cap c|}{|c|},$$

Metrics for Explainability (Cont.)

- Human evaluation
  - 30 volunteers evaluating the explainability of both taste clusters (Task 1) and user-item recommendations (Task 2)
  - Each volunteer is asked to look items' profiles and user's interactions
  - Then evaluate the results by comparison with baselines
  - Task 1
    - Rank the quality of generated clusters' tags
  - Task 2
    - Rank the quality of user-item explanation

Experimental Evaluation

Datasets

- Real-world datasets (Xbox) and public datasets (MovieLens and Last-FM)

Dataset	#Users	#Items	#Interactions	#Tags
Xbox	465,258	330	6,240,251	115
MovieLens	6,033	3,378	836,434	18
Last-FM	53,486	2,062	2,228,949	54

Recommendation performance

- Achieve excellent accuracy performance while providing interpretability
- Our method greatly outperforms the baseline in all metrics across all datasets

	Xbox				MovieLen				Last-FM			
	R@5	R@10	N@5	N@10	R@5	R@10	N@5	N@10	R@5	R@10	N@5	N@10
MF	0.2615	0.3686	0.2383	0.2824	0.0601	0.0975	0.2738	0.2511	0.0289	0.0446	0.0428	0.0443
NCF	0.2372	0.3433	0.2065	0.2503	0.0594	0.0985	0.2701	0.2517	0.0269	0.0456	0.0396	0.0383
CDAE	0.2604	0.3738	0.2346	0.2813	0.0609	0.0946	0.2671	0.2534	0.0286	0.0402	0.0431	0.0518
LightGCN	0.2684	0.3625	0.2382	0.2837	0.0699	0.1163	0.2979	0.2752	0.0398	0.0578	0.0605	0.0634
EFM	0.2647	0.3652	0.2368	0.2873	0.0657	0.1027	0.2866	0.2635	0.0319	0.0482	0.0471	0.0484
AMCF	0.2601	0.3613	0.2355	0.2806	0.0603	0.0986	0.2719	0.2498	0.0295	0.0488	0.0456	0.0457
MF_{forest}	0.2907	0.3983	0.2615	0.3159	0.0787	0.1276	0.3122	0.2911	0.0374	0.0548	0.0562	0.0594
ECF_{single}	0.1714	0.2763	0.1423	0.1854	0.0352	0.0608	0.1584	0.1505	0.0205	0.0315	0.0339	0.0345
ECF	**0.2970**†	**0.4299**†	**0.2644**†	**0.3193**†	**0.0788**	**0.1325**†	**0.3183**†	**0.2952**†	**0.0455**†	**0.0635**†	**0.0782**†	**0.0749**†

Experimental Evaluation

Explainability

- **K-means**: similarity-oriented method which utilizes item embedding from MF to perform K-means algorithm
- **TagCluster**: tag-oriented method which collects items with the same tags
- ECF takes all aspects into consideration so that it can avoid obvious shortcomings on a certain metric

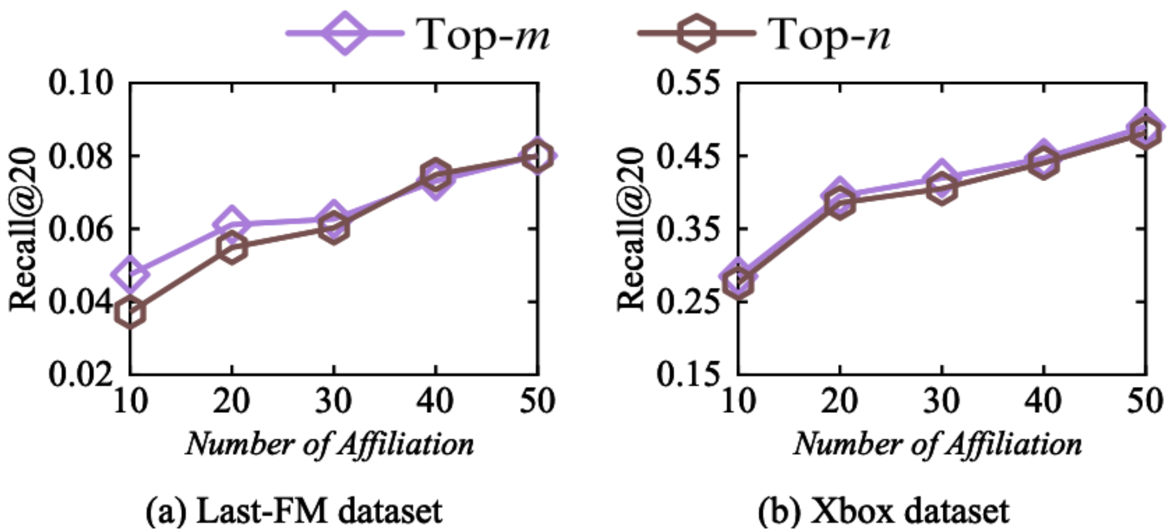
Method	Cov.	Util.	Sil.	Info.	Overall
Xbox					
ECF	0.8002	**0.7052**	0.2604	**0.3162**	**1.7463**
TagCluster	**0.9950**	0.2878	-0.1788	0.1579	0.9262
K-means	0.5710	0.3739	**0.4286**	0.0185	1.0563
Random	0.5396	0.1450	-0.3614	0.0125	0.0000
MovieLens					
ECF	0.7992	**0.7778**	0.1964	**0.3131**	**1.5651**
TagCluster	**0.991**	0.5259	-0.2573	0.1517	0.8898
K-means	0.6877	0.4478	**0.3265**	0.0168	0.9573
Random	0.5933	0.3672	-0.4452	0.0061	0.0000
Last-FM					
ECF	0.7648	**0.6259**	0.1584	**0.2996**	**1.5352**
TagCluster	**0.9880**	0.3703	-0.2511	0.1206	0.9143
K-means	0.5667	0.4841	**0.3197**	0.0182	1.0752
Random	0.5385	0.2275	-0.4673	0.0148	0.0000

Method	RankOfTask1	RankOfTask2
ECF	**1.73**	**1.3**
TagCluster	2	2.5
K-means	2.73	2.23
Random	3.63	3.93

Experimental Evaluation

Ablation Study

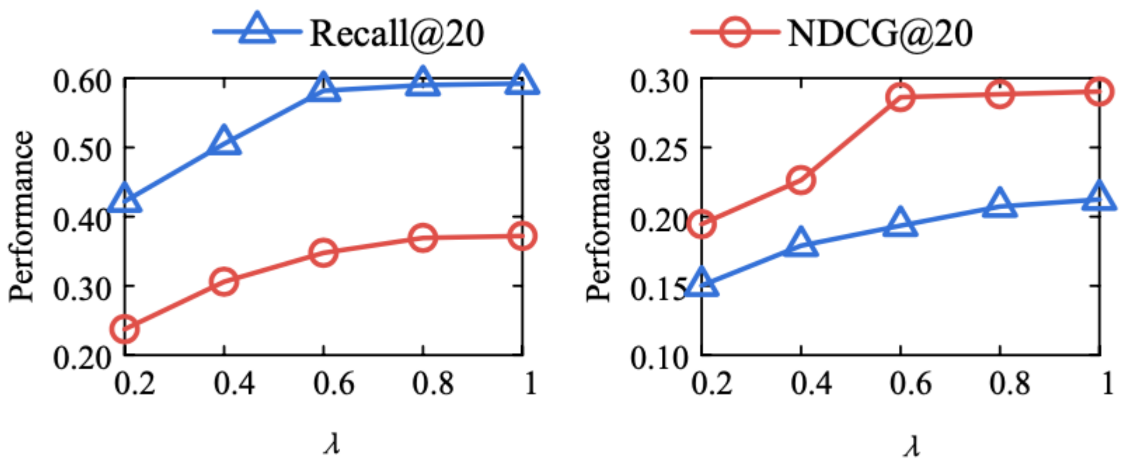
- Impact of top-m and top-n selection



Experimental Evaluation

Ablation Study

- Impact of the auxiliary collaborative signals λ

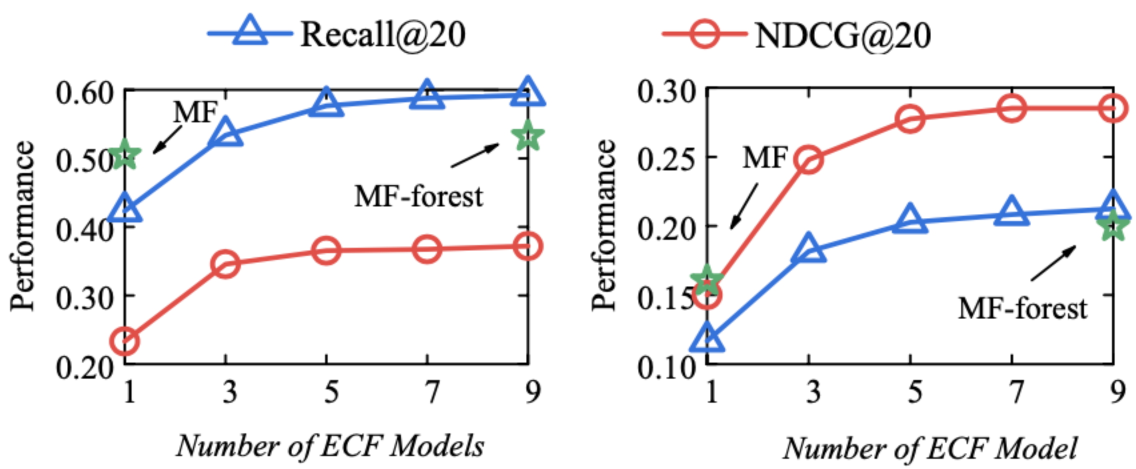


(a) Xbox dataset (b) MovieLens dataset

Experimental Evaluation

Ablation Study

- Impact of the forest mechanism



(a) Xbox dataset (b) MovieLens dataset

Case Study: Last-FM

Learned Taste Clusters

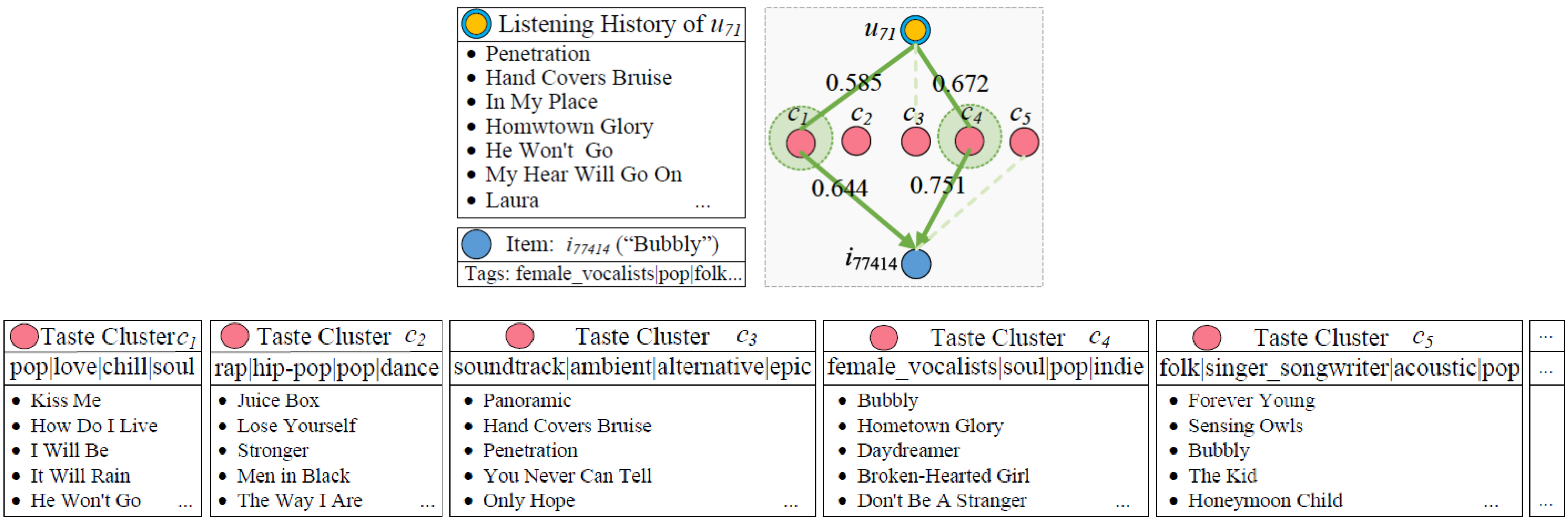
- Can be used to correct tags
  - Tags for *Bubbly*: ``female_vocalists|pop|folk|acoustic|love``
  - Missing tag ``singer_songwriter``
    - Colbie Caillat is also a songwriter who wrote the song

Taste Cluster c_1	Taste Cluster c_2	Taste Cluster c_3	Taste Cluster c_4	Taste Cluster c_5	...
pop|love|chill|soul	rap|hip-pop|pop|dance	soundtrack|ambient|alternative|epic	female_vocalists|soul|pop|indie	folk|singer_songwriter|acoustic|pop	...
• Kiss Me • How Do I Live • I Will Be • It Will Rain • He Won't Go ...	• Juice Box • Lose Yourself • Stronger • Men in Black • The Way I Are ...	• Panoramic • Hand Covers Bruise • Penetration • You Never Can Tell • Only Hope ...	• Bubbly • Hometown Glory • Daydreamer • Broken-Hearted Girl • Don't Be A Stranger ...	• Forever Young • Sensing Owls • $\boxed{\text{Bubbly}}$ • The Kid • Honeymoon Child ...	...

Case Study: Last-FM

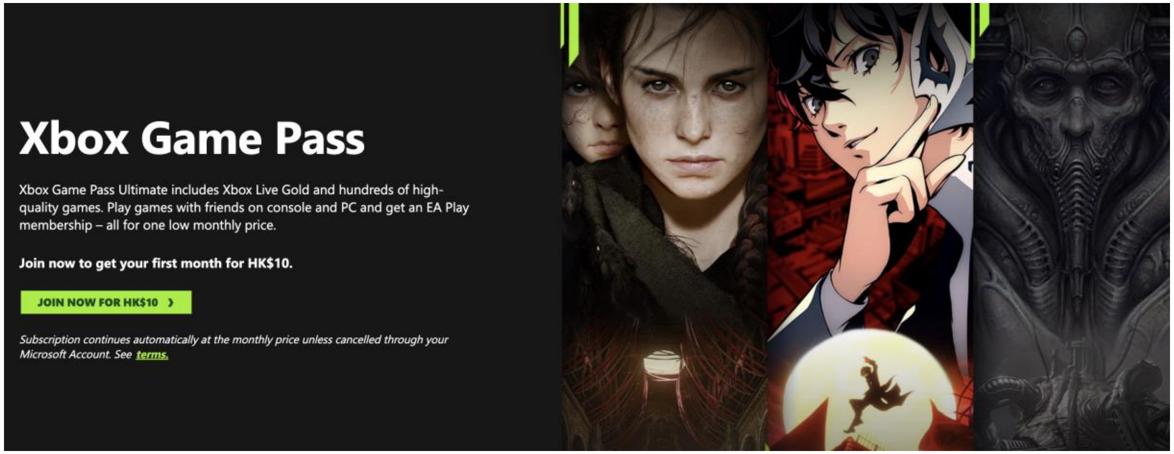
Explanations of the recommendation

- The weights of affiliation matrix indicate the relatedness between users/items with taste clusters
- Find the explanation paths for prediction score
  - $i_{71} \rightarrow c_1 \rightarrow i_{77414}$ and $i_{71} \rightarrow c_4 \rightarrow i_{77414}$



Case Study: Xbox Game Pass

- Real-world dataset with small items/games (~400)

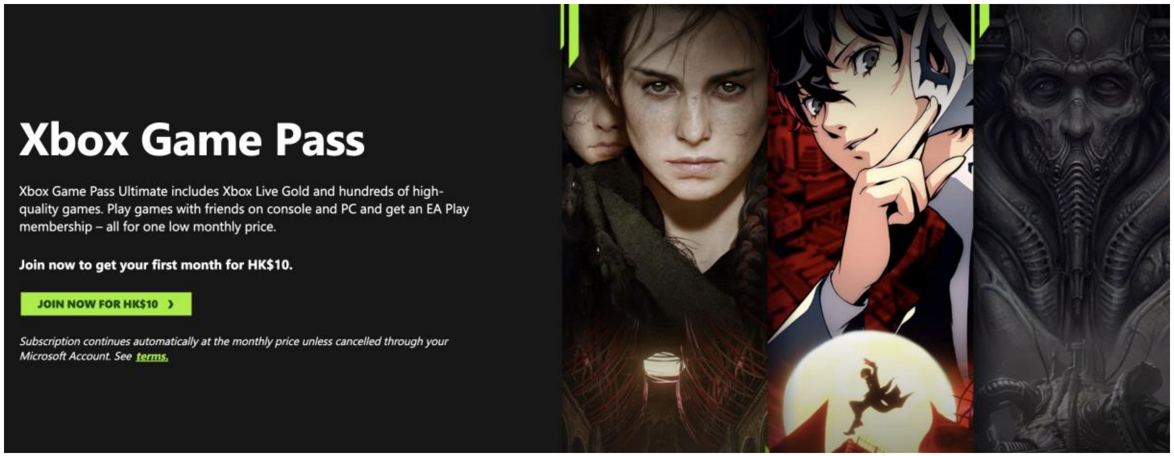


- Recommendation accuracy

model	Recall@5
Random	0.02
MF	0.18
ECF_{single}	0.14
ECF	**0.24**

Case Study: Xbox Game Pass

- Real-world dataset with small items/games (~400)



- Learned Taste Clusters

ClusterID	Tags	Hard Count	HalfHard Count	Count	In_sim	Cross_sim
36	Difficulty_Low| LearningCurve_Low| GoodForKids| MoodsMotivations_Cozy	6	10	15	0.595	0.003

Case Study: Xbox Game Pass

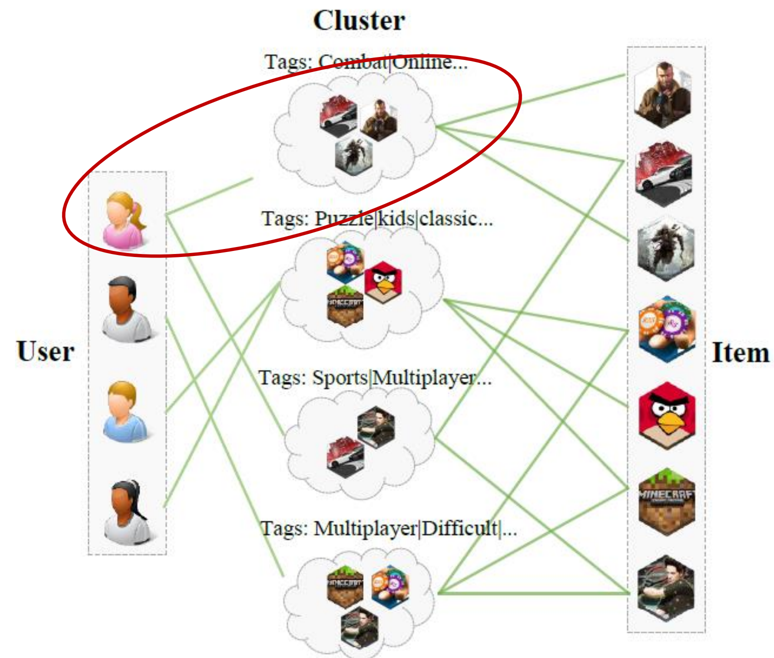
- Affiliated games

```
ClusterID,TitleID,Game,Hard
36,1970834532,RAGE 2 (PC),True
36,1891258006,The Bard's Tale ARPG : Remastered and Resnarkled,False
36,2102421406,Day of the Tentacle Remastered,False
36,2096242259,ANVIL : Vault Breaker (Game Preview),False
36,2096821037,DOOM 3,True
36,2013511408,DOOM Eternal Standard Edition (PC),True
36,1966816764,My Friend Pedro Win10,False
36,2008811924,Wolfenstein II: Standard Edition,False
36,1696012554,Goat Simulator Windows 10,False
36,1621285366,DOOM (1993),True
36,2055724194,SHENZHEN I/O,False
36,1843641391,Quake,True
36,2090892978,Farming Simulator 22,False
36,1805483741,Dishonored®: Death of the Outsider™ (PC),False
36,2078926688,DOOM II (Classic),True
36,1677025209,Wolfenstein: The New Order (PC),False
36,682562723,Halo: Spartan Assault,False
36,1822205071,The Anacrusis (Game Preview),False
36,1909396590,Wolfenstein: The Old Blood (PC),False
```

More Applications of ECF

Taste Cluster Recommendation

- A new but ubiquitous recommendation task like playlist recommendation in Spotify or gamelist recommendation in Xbox



User Profiling

- User-cluster affiliations discovered by ECF can also be used as user profiles directly
- Can be used for user-level predictive tasks, ad audience targeting and look-alike audience extension, etc.

Flexibility

- Applied with other popular embedding-based methods like LightGCN

	Performance		Explainability				
	R@20	N@20	Cov.	Util.	Sil.	Info.	Overall
ECF	0.0851	0.0773	0.7648	0.6259	0.1584	0.2996	1.5352
ECF_{LGN}	**0.0876**	**0.0792**	**0.7831**	**0.6430**	**0.1590**	**0.3042**	**1.5758**

Future direction

- Optimization
  - How to optimize taste clusters in an elegant way?
- Quality
  - How to tag taste clusters properly and improve the quality?
- Scalability
  - How to apply ECF to scenarios with millions of users and items?
- Generalization
  - Can we go beyond item tags? Knowledge graph, reviews…

Thank you !

Questions?

Yuntao Du
ytdu@zju.edu.cn