2020 IEEE International Conference on Data Mining (ICDM)

Interest Sustainability-Aware Recommender System

Dongmin Hyun[1], Junsu Cho[1], Chanyoung Park[2], Hwanjo Yu[1]
Dept. of Computer Science and Engineering, POSTECH, Pohang, Republic of Korea[1]
Dept. of Industrial and Systems Engineering, KAIST, Daejeon, Republic of Korea[2]
{dm.hyun, junsu7463, hwanjoyu}@postech.ac.kr, cy.park424@gmail.com

210520 Chia-Chun Ho

Outline

Introduction

Recommender System consider the concept drift

- According to Amazon report, about 30% their page views from recommendations
- An important aspect for building successful recommender system is to consider the **concept drift**.
- A user's interest changes over time, and the preference even towards the same type of items can change.
    - e.g. Most users who like wired earphones (e.g. EarPods) may change their interests over time and prefer wireless earphones (e.g. AirPods).

Introduction

Existing methods problem

- Existing methods capture the concept drift of users mainly based on each user's consumption history.
- Some sequential recommender systems take a user's N recently-consumed items as input to predict the next item that user would consume.
  - Represent the concept drift of the users via the order of items in a user's consumption history.
- Despite their success, previous sequential recommender systems are limited in that they <u>ignore how much users' interest in each item will sustain in the future</u>.

Introduction

Model the concept drift

- To model the concept drift of users, systems should focus on items that are likely to sustain users' interest until the deployment time (i.e. actual time at which items are recommend).

- Therefore, we should consider how likely each item is to sustain users' interest in deployment time.

- Supposed there are restaurants opened in 2013, where some restaurants have attracted users' interest until recently, while other restaurants have gradually lost users' interest.
- In this case, since the restaurants that belong to the former case are more likely to attract users in deployment time than those that belong to latter case.

Introduction

<u>C</u>ollaborative <u>R</u>epresentation Learning with <u>I</u>nterest <u>S</u>ustainability (CRIS)

- Take a totally different approach to model the concept drift of users.
- The key of this method is to recommend items based on the interest sustainability score (ISS), which is a score of how much users' interest in each item will sustain in the future.
- Prior to training the recommendation model, we first **<u>compute the ISS of each item by training a neural classifier in a supervised manner</u>**.
- Based on the predicted ISS of each item, the **<u>propose a metric learning framework to make users closer to the items with high ISSs in the representation space than those with low ISSs</u>**, thereby recommending items that would be attractive to users in the deployment time.

Introduction

Potential conflict between modeling the ISSs and the original objective of the metric learning

- For example, an item consumed by a user should be close to the user according to the original objective, but if the ISS of the item is low the item is forced to be distant from the user, which prevents the recommendation system from fully learning the user's preference for items.

- In the light of this issue, we further improve the method with prototypes to relieve the conflicts between the objectives.

Related Work
Recommender System

- General Recommender Systems
  - Matrix factorization (MF), Bayesian personalized ranking (BPR)
  - Collaborative Metric Learning (CML), Symmetric metric learning (SML)
- Sequential Recommender Systems
  - Caser (base on CNNs), Hierarchical gating network(HGN), SASRec (Self-attention)

Proposed Method

Collaborative Representation Learning with Interest Sustainability (CRIS)

- First demonstrate how to obtain the interest sustainability score (ISS) that quantifies how much users' interest in items will sustain in the future.
- Then based on the obtained ISS, we propose a metric learning framework for capturing the concept drift of users.

Proposed Method

Interest Sustainability Prediction

- Prior to training the recommender system, train a neural classifier, which predicts whether each item will be consumed in the future, to obtain the ISS for each item.

- Consider that we have user-item interaction data D such that:

  - $D = \{(u, i, t) \mid \text{user } u \text{ consumed item } i \text{ at time } t\}$

  - D: general source to train recommender systems.

Proposed Method

Interest Sustainability Prediction

- First divide D chronologically such that $D = D_f || D_b$.
- D_f , D_b denote the front, back part
- All interactions in D_f are precedent to any interaction in D_b.
- $||$ is concatenation operation.

- The divided data D_f and D_b are used for building input and labels:
  - $Input : i,$ item i that appears in D_f .
  - $Label : y_i = \begin{cases} 1, & \text{if } i \text{ appears in } D_b \, . \\ 0, & \text{otherwise} \, . \end{cases}$
- The goal is to predict whether item i, which appears in D_f , will be consumed in the future.

Proposed Method
Interest Sustainability Prediction

- Train a parameterized model M under a supervised-learning framework with binary cross entropy loss:
  - $$L_{IS} = \sum_{i}^{|I|} y_i log(M(f_i;\theta)) + (1 - y_i) log(1 - M(f_i;\theta))$$
  - θ: model parameters
  - f_i: feature representation of item i
- ISS is defined by the output of the trained model:
  - $p_i = M(f_i;\theta)$
  - $p_i \in \mathbb{R}$: ISS of item i in the form of probability.

Proposed Method

Interest Sustainability Prediction - Predictive Model and Feature

- Given the classification problem, introduce f_i, M as shown in Fig.2.
- Intuitively, the consumption pattern of an item over time will be an important clue in determine consumed in the feature.
- To model the consumption patterns of items over time, we represent the timestamps at which an item was consumed as frequency bins:
  - item : $[t_1, t_2, \cdots, t_N] \xrightarrow{Binning} [b_1, b_2, \cdots, b_B]$
  - t_j : j-th timestamp at which an item was consumed
  - N: number of consumptions of the item in D_f
  - b_k: number of times an item was consumed in the period of k-th frequency bin
  - B: number of bins where $N \gg B$

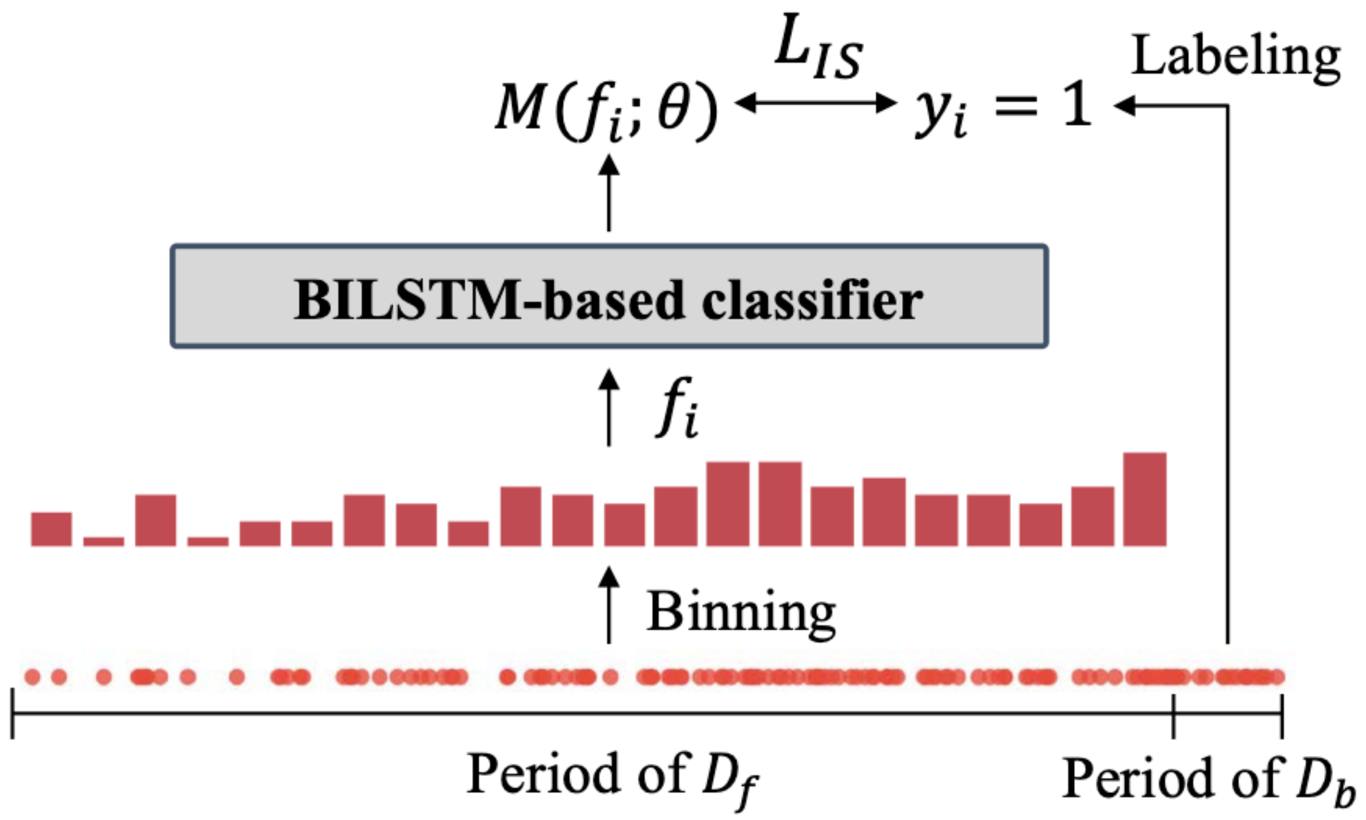
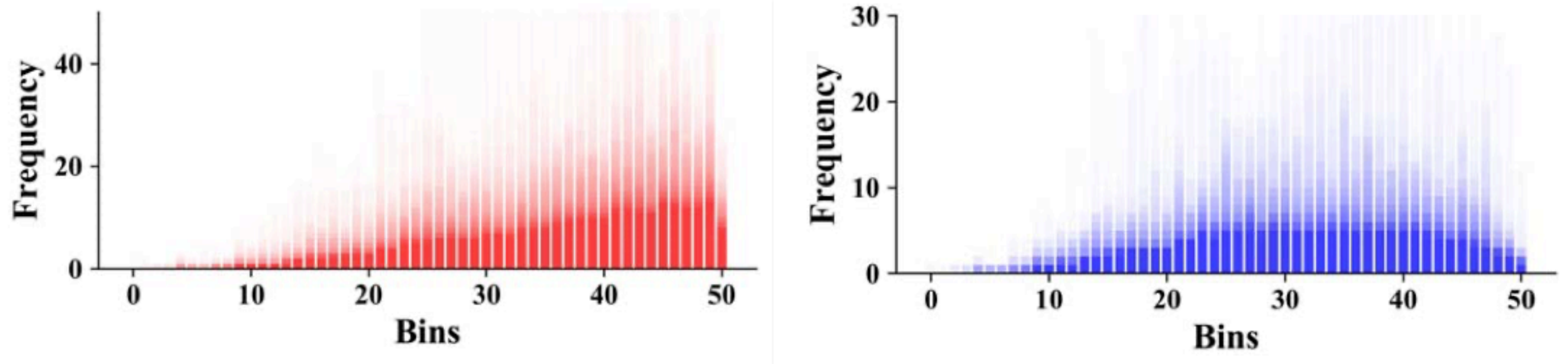


Fig. 2: Training process of a propose classifier on the interest sustainability prediction.

Proposed Method

Interest Sustainability Prediction - Predictive Model and Feature

- To examine the benefit of the frequency bins, Fig.3 show the distribution of the frequency bins that belong to $y_i = 1$ or $y_i = 0$.
- Observe that the values in the frequency bins:
  - $y_i = 1$ tend to gradually increase over time
  - $y_i = 0$ tend to decrease in recent periods



(a) $y_i = 1$ (b) $y_i = 0$

Fig. 3: Distribution of frequency bins corresponding to 10,000 randomly-sampled items that belong to $y_i = 1$ (a) or $y_i = 0$ (b) on Yelp dataset.

- Therefore, use the features that capture the consumption patterns changing over time (sequence of frequency bins) to predict items will be consumed in the future.

Proposed Method

Interest Sustainability Prediction - Predictive Model and Feature

- Based on the frequency bins, design a RNN as a sequence encoder, adopt BILSTM, which has been effective to model sequential data.
- Design the predictive model with BILSTM as follows:
  - $M(f_i;\theta) = \sigma(\mathbf{w}^\top(\overrightarrow{LSTM}(f_i) \| \overleftarrow{LSTM}(f_i)) + c)$
  - $f_x = [b_1, b_2, \cdots, b_B] \in \mathbb{R}^B$: sequence of frequency bins of item i
  - σ: sigmoid function, $\mathbf{w} \in \mathbb{R}^{2l}$: trainable weight, $\mathbf{c} \in \mathbb{R}$: bias

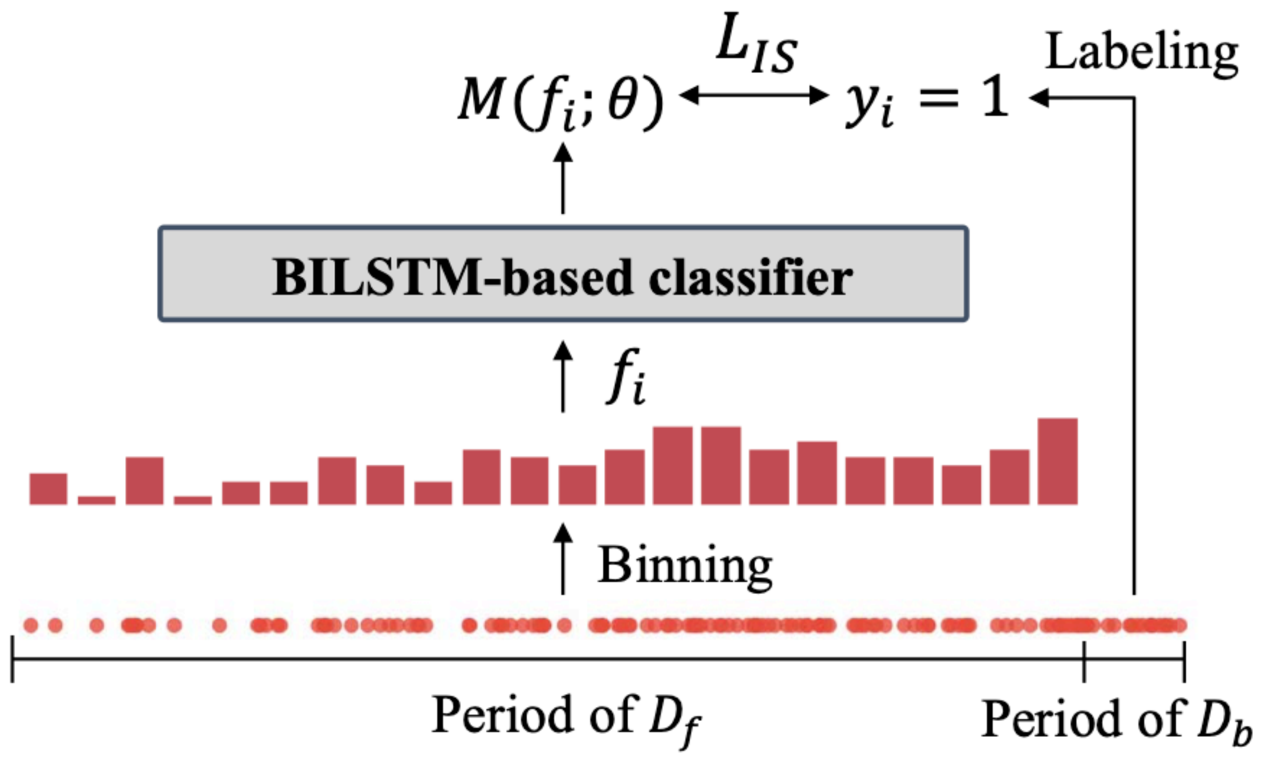
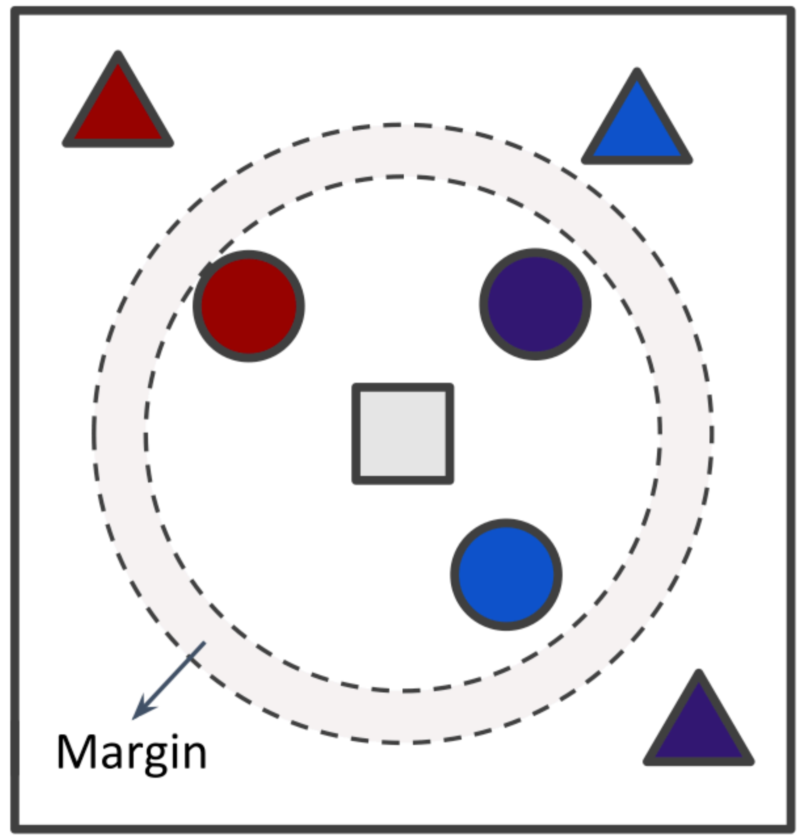


Fig. 2: Training process of a propose classifier on the interest sustainability prediction.

- Each LSTM encodes the feature f_i into l-dimentional vector, which obtained from their last hidden state.

Proposed Method

Metric Learning with Interest Sustainability Score



(a) CML

- Task: recommendation with the ISS p_i to model how users' interest in each item will sustain in the future.

- The basis of the proposed recommender system is a metric learning framework, which makes users closer to items consumed by them (positive items) than items not consumed by them (negative items) as shown in Fig.4a.

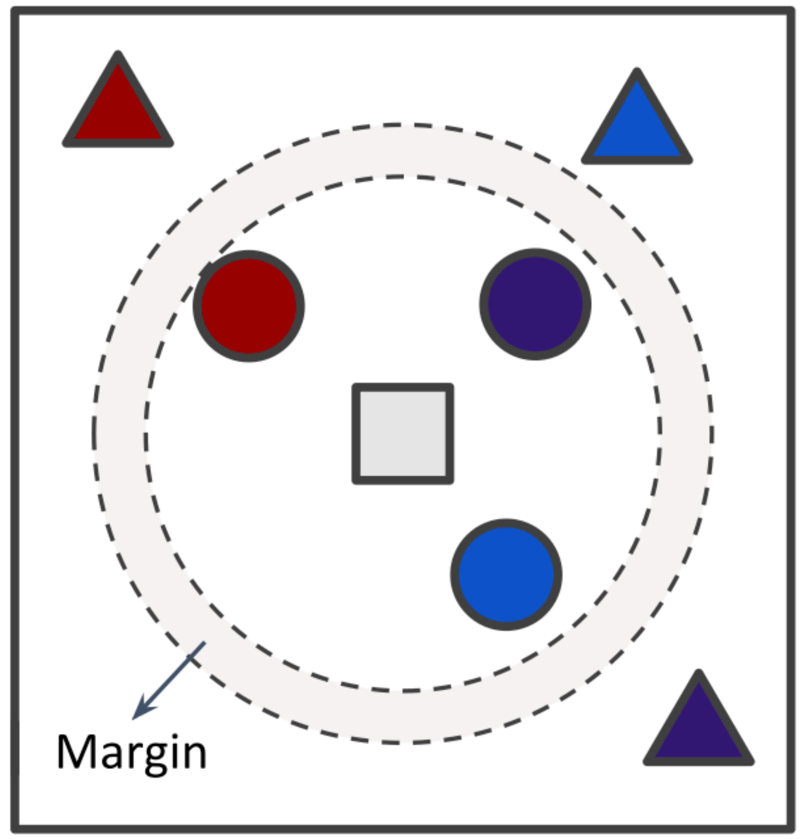
- Consumption-based objective L_C defined as follows:

  - $L_C(u, i^+, i^-) = [m + d(\mathbf{u}, \mathbf{i}^+) - d(\mathbf{u}, \mathbf{i}^-)]_+$

  - $[x]_+ = \max(x,0)$, $\mathbf{u}, \mathbf{i}^+, \mathbf{i}^- \in \mathbb{R}^K$: embedding vectors of user, positive / negative item

Proposed Method

Metric Learning with Interest Sustainability Score

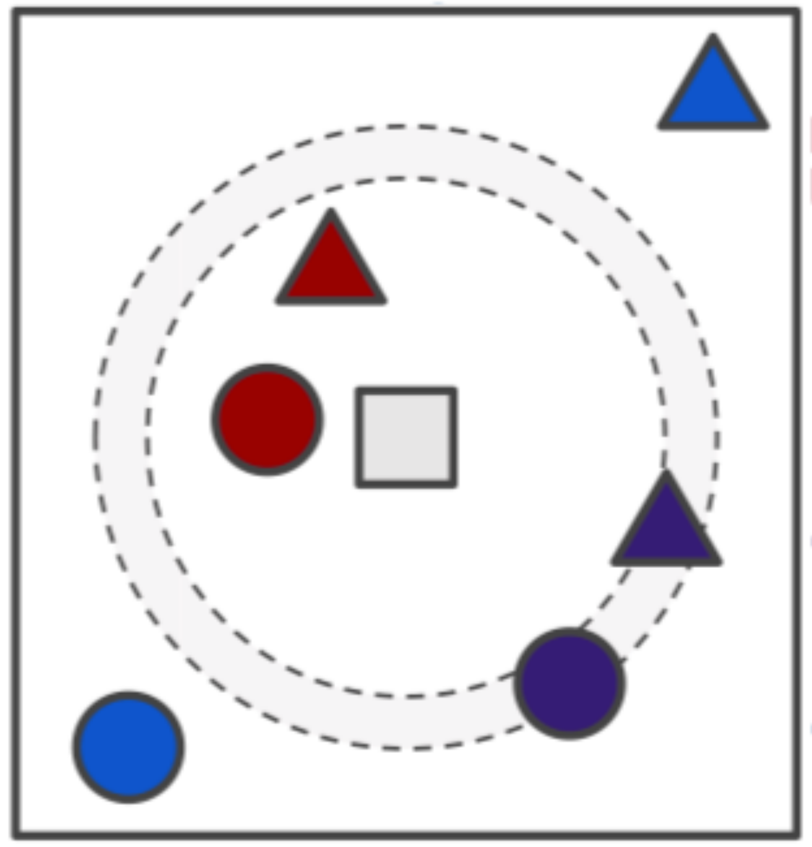


(a) CML

- $L_C(u, i^+, i^-) = [m + d(\mathbf{u}, \mathbf{i}^+) - d(\mathbf{u}, \mathbf{i}^-)]_+$
- Used euclidean distance as a distance metric d.
- Margin $m \in \mathbb{R}_{>0}$ imposes u to be closer to i^+ than i^- by m in the representation space.
- Impose the space to be a unit sphere by normalizing the embedding vectors (e.g., $\mathbf{u} \leftarrow \mathbf{u}/\max(1, \|\mathbf{u}\|^2)$) for each epoch.

Proposed Method

Metric Learning with Interest Sustainability Score



(b) CRIS^{reg}

- Incorporate the ISS in the above metric learning framework to consider how users' interest in each item will sustain in the future.

- The underlying idea is pull items with high ISS to users and to push items with low ISS from users.

- Design a ISS-based objective L_S with continuous labels (p_i):

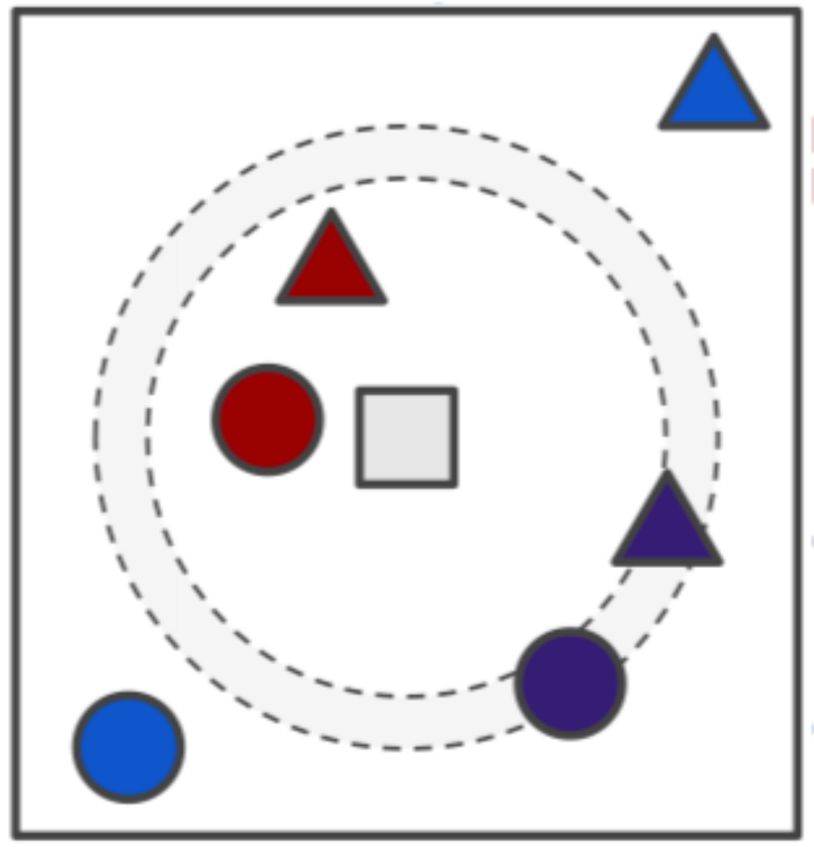
  - $L_S(u, i^+, i^-) = \{(d(\mathbf{u}, \mathbf{i}^+) - d(\mathbf{u}, \mathbf{i}^-)) - (p_{i^-} - p_{i^+})\}^2$

- The goal of L_S is to arrange item i^+ and i^- by according to the difference of their ISSs ($p_{i^-} - p_{i^+}$).

- For example, if $p_{i^-} - p_{i^+} < 0$, the objective makes the positive item will be closer to the user than the negative item by $|p_{i^-} - p_{i^+}|$.

Proposed Method

Metric Learning with Interest Sustainability Score



(b) CRIS^{reg}

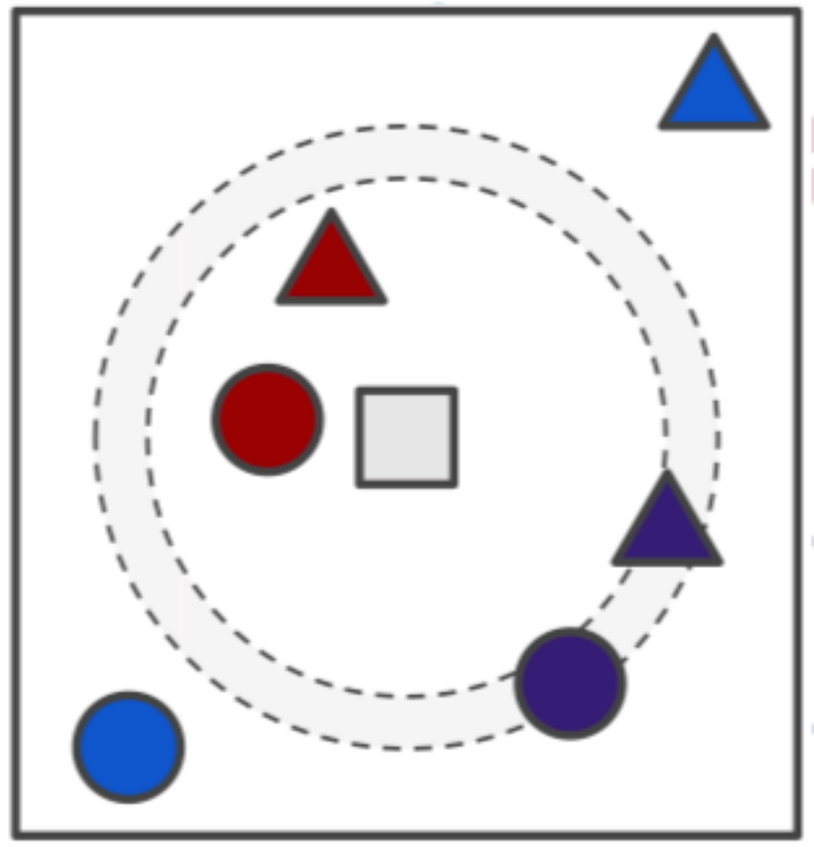
- The final loss is a linear combination of both objectives:

- $$L = \sum_{(u,i^+)\in P} \sum_{(u,i^-)\notin P} L_C(u, i^+, i^-) + \lambda L_S(u, i^+, i^-)$$

  - P: set of user-item interactions, λ: balancing coefficient,
    L_S: regularization on metric learning framework.

- Given the combination of both objectives, the metric learning method can build a representation space with considering both whether users liked items (by L_C) and how users' interest in the items sustain in the future (by L_S), name this method as CRIS^{reg}

Proposed Method
Prototype Learning


(b) CRIS^{reg}

- A limitation in the metric learning framework with the ISS, there can be potential conflicts between two objectives because an anchor (i.e. a user) is shared to optimize both objectives, L_C and L_S.

- For example, positive item of a user can have low ISS, thus consequently the positive item can be distant from the user. (Fig.4b) Therefore, modeling the ISS can prevent the recommendation system from fully learning the user's preference for items.

Proposed Method

Prototype Learning

- To alleviate such conflicts, further extend the metric learning framework with prototypes, which are trainable points in the representation space.
- In this work, design each prototype to be responsible for optimizing one objective. The intuition is to disentangle two objectives by using two types of anchors (prototypes) instead of a single type of anchors (users).

Proposed Method

Prototype Learning

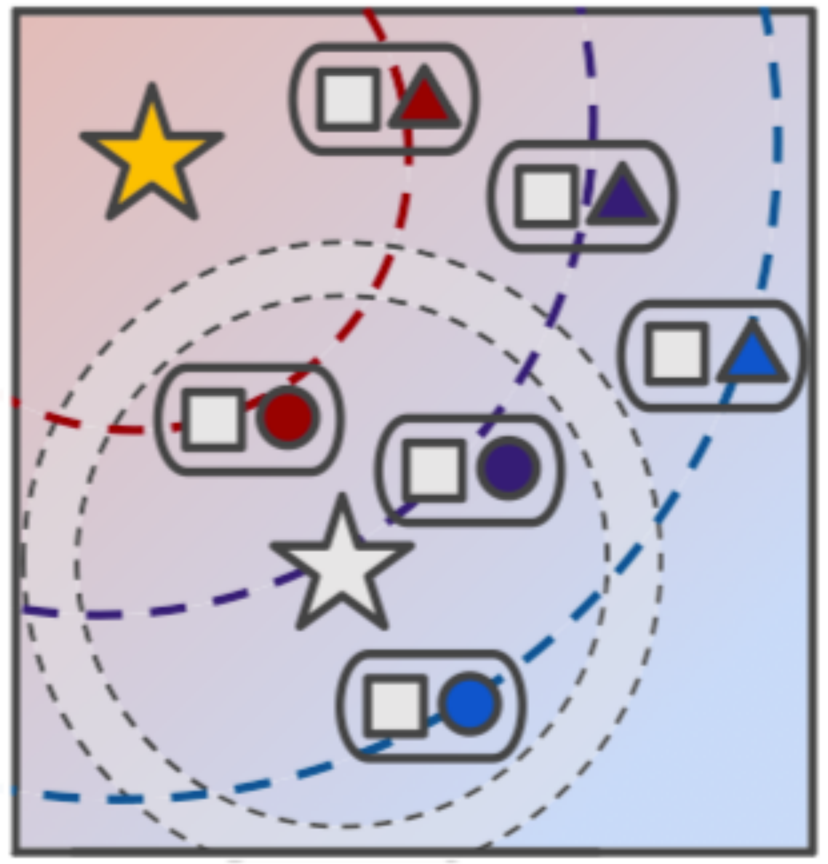
- First define two prototypes in the representation space: $\mathrm{C}, \mathrm{S} \in \mathbb{R}^K$
  - C is a prototype for optimizing the consumption objective L_C
  - S is a prototype for optimizing the interest sustainability objective L_S
- Then project a user-item pair into a single point such that: $\mathrm{T}_{u,i} = \mathbf{u} + \mathbf{i}$
  - T is a transformation function and use sum operation.

Proposed Method

Prototype Learning

- Given two prototypes, reformulate the objectives of CRIS^{reg} as follows:
  - $L_C^P(u, i^+, i^-) = [m + d(\mathsf{C}, \mathsf{T}_{u,i^+}) - d(\mathsf{C}, \mathsf{T}_{u,i^-})]_+$
  - $L_S^P(u, i^+, i^-) = \{(d(\mathsf{S}, \mathsf{T}_{u,i^+}) - d(\mathsf{S}, \mathsf{T}_{u,i^-})) - (p_{i^-} - p_{i^+})\}^2$
- Based on the prototypes, the consumption loss L_C^P makes the pair of a user and T_{u,i^+} closer to prototype C than the pair of the user and T_{u,i^-}.
- Similarly, L_S^P make the pair of a user and an item with higher ISS closers to prototype S than user and an item with lower ISS.

Proposed Method
Prototype Learning



(c) CRIS

- The recommender system can optimize both objectives with less conflicts between them than the approach of CRIS^{reg}.

- Combine the prototype-based objectives with a balancing coefficient λ:

  - $$L^P(\theta) = \sum_{(u,i^+)\in P} \sum_{(u,i^-)\notin P} L_C^P(u,i^+,i^-) + \lambda L_S^P(u,i^+,i^-)$$

- Train the system by minimizing the loss using SGD with respect to the θ (i.e. $\min_\theta L^P(\theta)$)

- Under the prototype-based learning, a recommendation score of user u on item i is as follow:

  - $Score(u,i) = -\{d(\mathrm{C}, \mathrm{T}_{u,i}) + \gamma d(\mathrm{S}, \mathrm{T}_{u,i})\}$, γ: parameter to control the importance of the ISS

Experiments

Dataset

- Amazon, Yelp, GoodReads
- Filtered out noisy data from Yelp and GoodReads datasets by maintaining only user who made at least 10 interactions and item that were involved to at least 5 interactions.

TABLE I: Data Statistics. Int. denotes user-item interactions.

Data	# Users	# Items	# Int.(M)	Avg. Int. per user	Period
Tools	16,472	10,177	0.133	7.7	Nov 1999 - Jul 2014
Toys	19,153	11,865	0.165	8.3	Jul 2000 - Jul 2014
Cell Phones	27,372	10,279	0.190	6.5	Feb 2001 - Jul 2014
Clothing	38,651	22,974	0.274	6.6	Mar 2003 - Jul 2014
Sports	34,974	18,294	0.291	7.9	Mar 2002 - Jul 2014
Health	37,842	18,358	0.339	8.4	Dec 2000 - Jul 2014
Kindle	67,193	58,110	0.935	12.7	Mar 2000 - Jul 2014
CDs	74,926	64,342	1.093	14.4	Nov 1997 - Jul 2014
Movies	122,923	49,976	1.688	13.3	Nov 1997 - Jul 2014
Yelp	47,906	78,734	2.304	47.2	Oct 2004 - Nov 2018
GoodReads	58,003	45,330	2.791	47.5	Feb 2001 - Nov 2017

Experiments

Evaluation Protocol

- Used user-item interactions in the least one month as test data, and the others as training data, then set the interactions in the latest one month in the training data as validation data.
- Removed users and items, which don't appear in the training data, from the validation and test data.
- As metrics, adopt hit ratio (H$@k$) and normalized discounted cumulative gain (N$@k$) to evaluate the ranking performance.
- Due to space limitation, report results with $k = 10$, ran system 5 times and averaged results.

Experiments
Methods Compared

- CML (metric learning method): models users' preference to items with a metric instead of inner product.
- SML (metric learning method): state-of-the-art method enhancing CML by including a item-centric metric and trainable margins.
- NTF: utilizes timestamp of interaction to capture user's periodical behaviors by extending tensor factorization with a neural network.
- Caser: sequential recommender system based on CNNs to extract local features from the sequence of users' consumption.
- SASRec: consider pair-wise interaction between items in the sequence of users' consumption via self-attention mechanism.
- TiSARec: extends SASRec by exploiting the time interval between two consecutive in the sequence of users' consumption.
- HGN: state-of-the-art sequential recommender system based on a hierarchical gating network to capture long/short term interaction
- CRIS^{reg}
- CRIS$^{wt.}$: straightforward method to utilize the ISS with the learning method. → $Score(u, i) = (p_i)^{\lambda} \cdot (-d(u, i))$
- CRIS

Experiments

TABLE II: Performance comparison. ΔH and ΔS are the relative improvements (%) of CRIS over HGN and SML, respectively, with the statistical significance p < 0.001 computed using the paired t-test.

Recommendation Performance Analysis-Performance Comparison with Baseline Methods

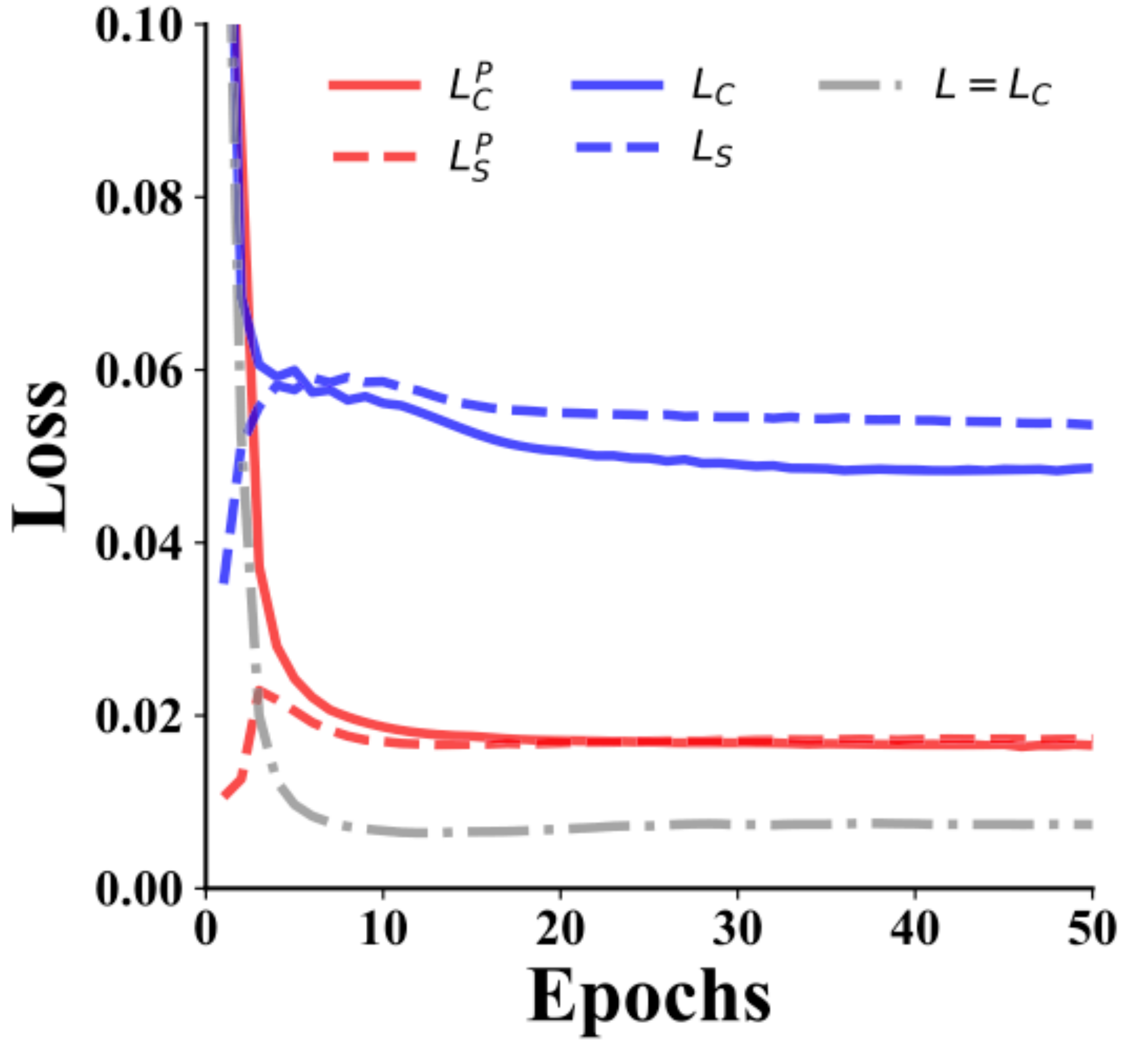
Dataset	Metric	BPR	CML	SML	NTF	Caser	SASRec	TiSASRec	HGN	CRIS^{reg}	$\text{CRIS}^{wt.}$	CRIS	Δ_H	Δ_S
Tools	H@10	0.3314	0.3649	0.3740	0.3449	0.3301	0.3044	0.3264	0.3605	0.3804	0.3953	**0.4047**	12.3	8.2
	N@10	0.1818	0.2009	0.2016	0.1951	0.1858	0.1660	0.1795	0.2061	0.2118	0.2190	**0.2276**	10.4	12.9
Toys	H@10	0.3586	0.3881	0.3906	0.3496	0.3426	0.3352	0.3352	0.3848	0.4275	0.4586	**0.4602**	19.6	17.8
	N@10	0.2155	0.2306	0.2343	0.197	0.1924	0.1870	0.1831	0.2269	0.2561	0.2656	**0.2726**	20.1	16.3
Cell Phones	H@10	0.4278	0.4547	0.4709	0.5315	0.4711	0.4659	0.4793	0.4763	0.5300	0.4620	**0.5642**	18.5	19.8
	N@10	0.2675	0.2825	0.2901	0.3190	0.2899	0.2790	0.2930	0.3037	0.3203	0.2863	**0.3416**	12.5	17.8
Clothing	H@10	0.3657	0.4073	0.4121	0.3809	0.3443	0.3421	0.3340	0.3912	0.4254	0.4016	**0.4473**	14.3	8.5
	N@10	0.2149	0.2437	0.2443	0.2117	0.1990	0.1959	0.1878	0.2339	0.2511	0.2394	**0.2652**	13.4	8.6
Sports	H@10	0.4458	0.4909	0.4914	0.4256	0.4366	0.4250	0.4216	0.4659	0.4877	0.4857	**0.5171**	11.0	5.2
	N@10	0.2637	0.2891	0.2887	0.2433	0.2566	0.2469	0.2430	0.2823	0.2853	0.2878	**0.3056**	8.3	5.9
Health	H@10	0.4239	0.4713	0.4746	0.4431	0.4336	0.4272	0.4396	0.4586	0.4804	0.4728	**0.4985**	8.7	5.0
	N@10	0.2501	0.2843	0.2835	0.2717	0.2639	0.2487	0.2632	0.2972	0.2972	0.2825	**0.3056**	2.8	7.8
Kindle	H@10	0.7136	0.7235	0.7235	0.5945	0.6403	0.6082	0.6497	0.7083	0.7603	0.7214	**0.7871**	11.1	8.8
	N@10	0.4672	0.4829	0.4834	0.3541	0.4019	0.3748	0.4141	0.4759	0.5171	0.4805	**0.5462**	14.8	13.0
CDs	H@10	0.6959	0.7104	0.7046	0.6426	0.5815	0.5826	0.6107	0.6591	0.7189	0.6727	**0.7389**	12.1	4.9
	N@10	0.4470	0.4610	0.4585	0.4003	0.3513	0.3563	0.3764	0.4289	0.4782	0.4404	**0.4931**	15.0	7.5
Movies	H@10	0.6938	0.7024	0.7020	0.6785	0.6421	0.6597	0.6553	0.6771	0.7056	0.6951	**0.7250**	7.1	3.3
	N@10	0.4504	0.4543	0.4544	0.4428	0.4111	0.4234	0.4244	0.4549	0.4582	0.4570	**0.4686**	3.0	3.1
Yelp	H@10	0.8715	0.8853	0.8857	0.8348	0.8052	0.8383	0.8701	0.8658	0.8928	0.8861	**0.9070**	4.8	2.4
	N@10	0.6031	0.6305	0.6294	0.5578	0.5146	0.5503	0.5829	0.5969	0.6138	0.6300	**0.6630**	11.1	5.3
GoodReads	H@10	0.7442	0.7541	0.7518	0.7243	0.6997	0.6437	0.7219	0.7381	0.7559	0.7576	**0.7920**	7.3	5.3
	N@10	0.5005	0.5115	0.5105	0.4906	0.4892	0.4293	0.5067	0.5308	0.5032	0.5144	**0.5377**	1.3	5.3

Experiments

Recommendation Performance Analysis-Comparison with Variants of CRIS

- Training convergence for each loss in CRIS and CRIS^{reg} along with consumption-based loss ($L = L_C$) that is optimized without the ISS-based loss (CML).

- We can observe that losses of CRIS converge at a lower point that those of CRIS^{reg}, which means the prototypes are indeed helpful to reduce the conflict between L_C and L_S.

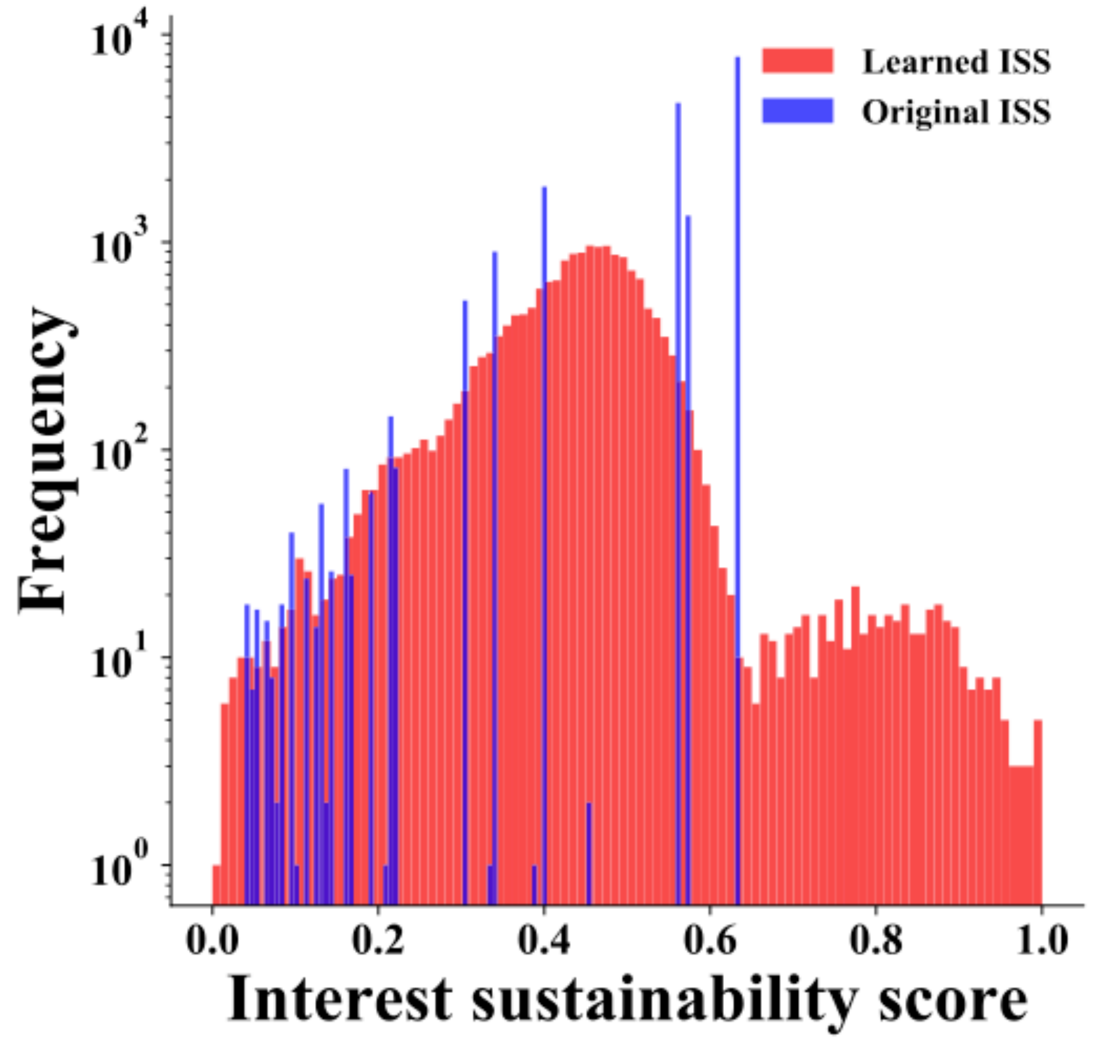


(a) Convergence comparison of CRIS, CRIS^{reg}, and CML.

Experiments

Recommendation Performance Analysis-Comparison with Variants of CRIS

- In Fig(b), compare the original ISSs (p_i) and the ISSs learned by CRIS ($\hat{p}_i$).
- To obtain $\hat{p}_i$, first compute the distance $d(\mathrm{S}, \mathrm{T}_{u,i})$ between the S (interest-sustainability prototype) and $\mathrm{T}_{u,i}$ (all pairs of users and items).
- Then average the distances for each item and take min-max normalization on the averaged distances for items to ensure within $x_i \in [0,1]$.
- Lastly, take the complement of the values ($\hat{p}_i \leftarrow 1 - x_i$) to obtain the ISSs learned by CRIS tend to follow the original ISSs, but smoother than the original.
- Conjecture the original ISSs can be inaccurate, thus consequently CRIS learns to reduce the noise. Therefore, CRIS is more robust to the the noise of the ISSs than the CRIS$^{wt.}$.



(b) Histogram of original and learned ISSs.

Experiments
Ablation Study

Dataset	Metric	Only $\mathcal{C}$	Only $\mathcal{S}$	Rand.	Oracle	NN	CRIS
Toys	H@10	0.385	0.352	0.361	0.652	0.401	0.460
	N@10	0.233	0.175	0.199	0.377	0.219	0.273
Clothing	H@10	0.401	0.320	0.371	0.543	0.291	0.447
	N@10	0.240	0.157	0.207	0.309	0.158	0.265
Health	H@10	0.460	0.237	0.411	0.538	0.414	0.499
	N@10	0.279	0.132	0.249	0.325	0.235	0.306
Movies	H@10	0.705	0.554	0.678	0.793	0.690	0.725
	N@10	0.458	0.299	0.446	0.518	0.433	0.469
Yelp	H@10	0.887	0.429	0.890	0.961	0.879	0.906
	N@10	0.632	0.193	0.629	0.738	0.617	0.657

- Only C is trained only the consumption-based loss L_C^P, shows similar performance of CML.
- Only S is trained only on the ISS-based loss L_S^P, shows the consistent degradation in the accuracy of recommendations.
- This result indicates that should utilize the ISSs along with the consumption-based objective, since modeling the ISS alone can't capture user's personalized preferences.
- Randomly ISSs (Rand.) hurt the model performance in all the cases as the representations of users and items will be wrongly learned by random ISSs.
- As oracle, set ISS p_i as 1 if item i appear in the test data and 0 else. It's provides the upper bound of the performance of CRIS.

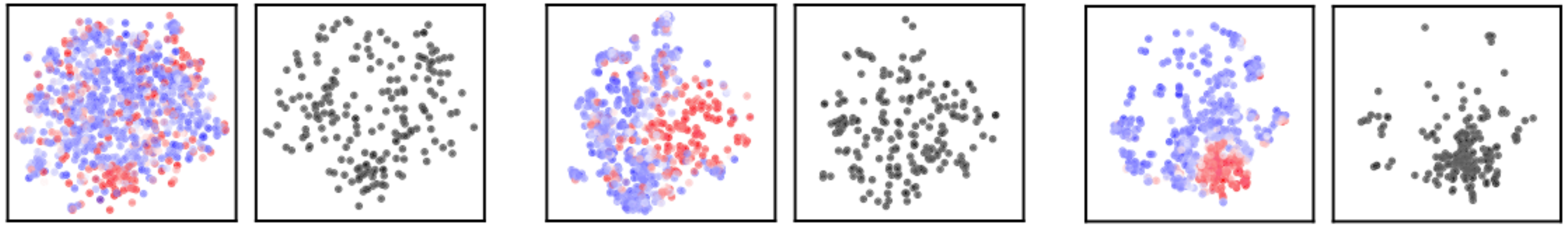
Experiments
Ablation Study

Dataset	Metric	Only $\mathcal{C}$	Only $\mathcal{S}$	Rand.	Oracle	NN	CRIS
Toys	H@10	0.385	0.352	0.361	0.652	0.401	0.460
	N@10	0.233	0.175	0.199	0.377	0.219	0.273
Clothing	H@10	0.401	0.320	0.371	0.543	0.291	0.447
	N@10	0.240	0.157	0.207	0.309	0.158	0.265
Health	H@10	0.460	0.237	0.411	0.538	0.414	0.499
	N@10	0.279	0.132	0.249	0.325	0.235	0.306
Movies	H@10	0.705	0.554	0.678	0.793	0.690	0.725
	N@10	0.458	0.299	0.446	0.518	0.433	0.469
Yelp	H@10	0.887	0.429	0.890	0.961	0.879	0.906
	N@10	0.632	0.193	0.629	0.738	0.617	0.657

- Examine a neural transformation (fully-connect layer) to project (user, item) into a point in the space instead of the sum operation.
  - On all datasets, the neural transformation is worse than the sum-based transformation.
  - Suspect the additional parameters of the NN make the method to overfit compared to the parameter-free approach (i.e. sum).

Experiments

Comparison of Learned Representations



(a) HGN (b) SML (c) CRIS

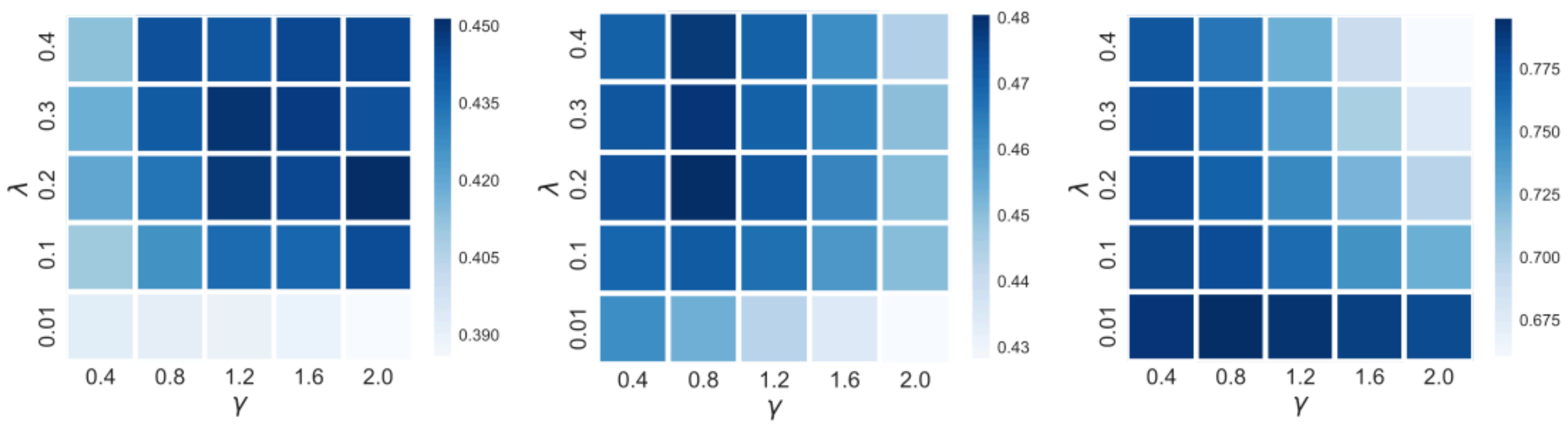
- Visualize the item representations learned by SML,HGN, CRIS to investigate whether they can consider the interest sustainability of items.
- Observe the baseline methods (HGN, SML) suffer from distinguishing the item with respect to their ISSs. Thus, items that appear over the representation space in test time.
- This result indicates that modeling only user-item interactions is limited to capture whether each item will be consumed in the future.
- CRIS successfully captures the interest sustainability, appear more clearly clustered that those baselines.
- Thus, we conclude that ISSs of items are essential signal that enables system to consider how users' interest will sustain in the future.

Experiments

$$L^P(\theta) = \sum_{(u,i^+)\in P} \sum_{(u,i^-)\notin P} L_C^P(u,i^+,i^-) + \lambda L_S^P(u,i^+,i^-)$$

$$Score(u,i) = -\{d(\mathsf{C}, \mathsf{T}_{u,i}) + \gamma d(\mathsf{S}, \mathsf{T}_{u,i})\}$$

Effect of Balancing Coefficients



(a) Toys (b) Health (c) Yelp

- Illustrates the sensitivity of the balance coefficients λ and γ:
  - CRIS achieves the best performance with small λ but large γ, which indicates the importance of the ISSs.
  - Conjecture the inconsistency between training and evaluation time is caused by the noise in the ISSs.
  - While training, CRIS depends less on the L_S^P to avoid overfitting to noisy ISSs.
  - In the evaluation time, CRIS largely depends on the denoised ISSs when determining the recommendation scores.
- CRIS can handle the noise in the ISSs by adjusting the balance coefficients λ and γ
- λ less than 0.5 is the best, reaffirms that ISSs should modeled with L_C^P to learn users' personalized preference.

Experiments

Effect of Periods

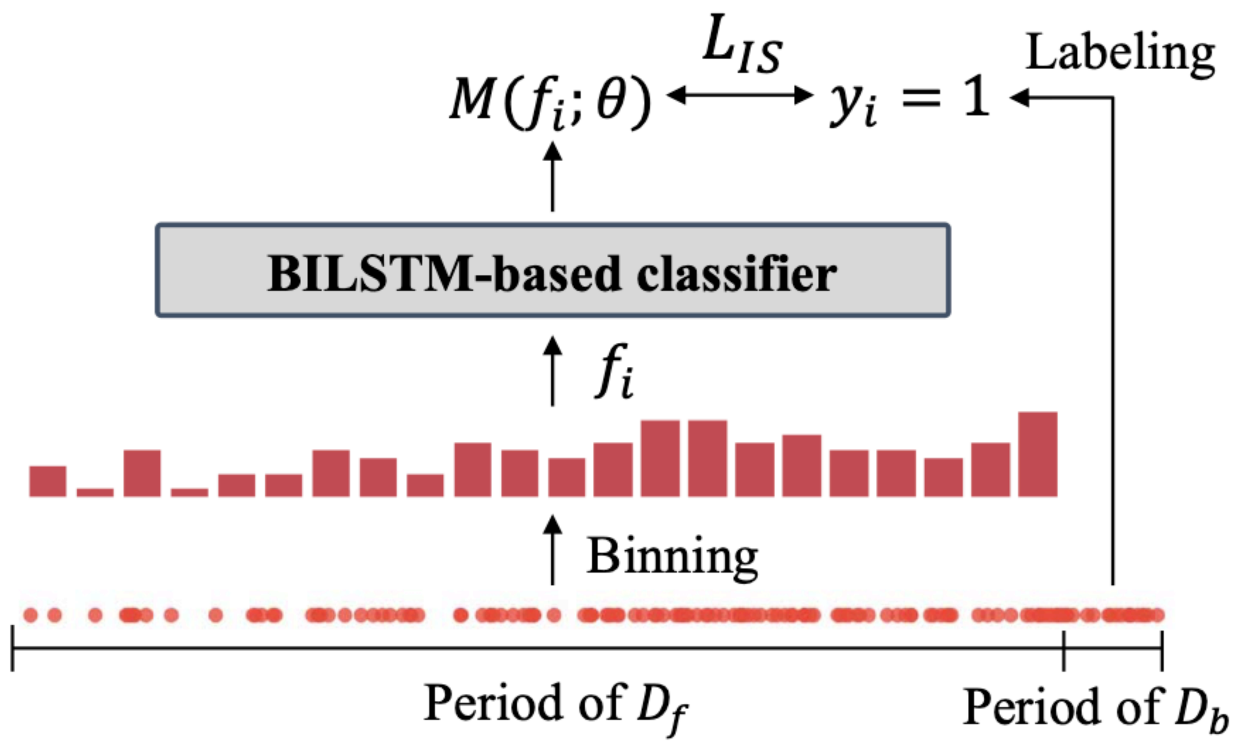
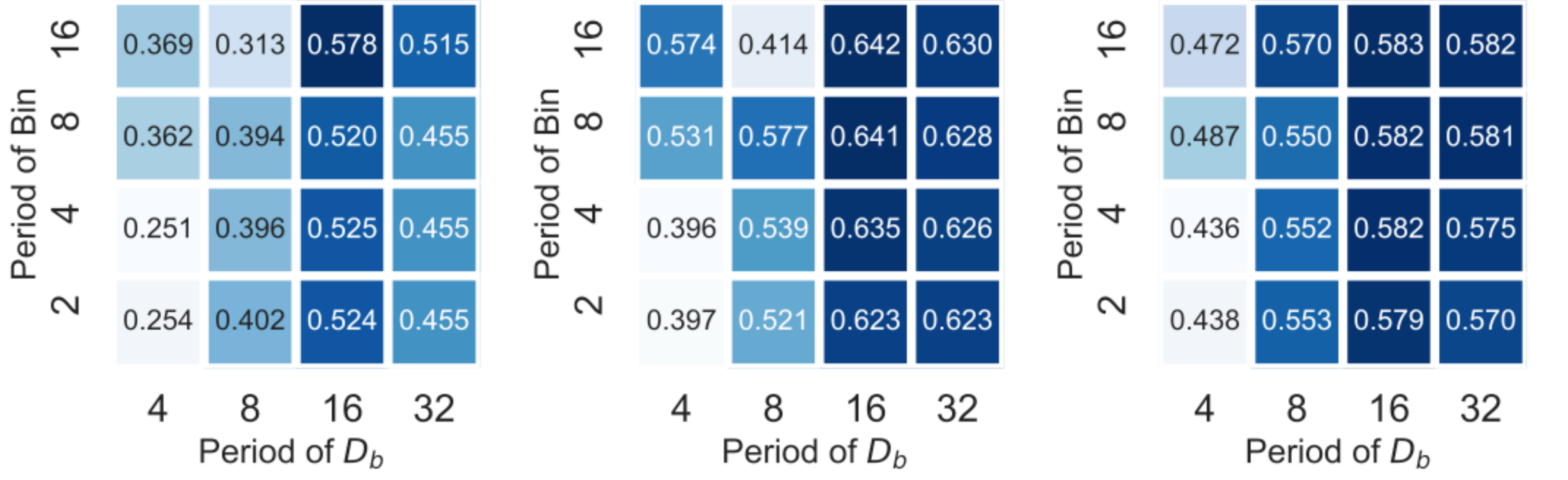


Fig. 2: Training process of a propose classifier on the interest sustainability prediction.



(a) Clothing (b) Kindle (c) GoodReads

Fig. 8: Sensitivity analysis on the periods of data D_b and frequency bins. The numbers in both axes denote the number of weeks.

- Performances are sensitive to the period of D_b, and long periods show the best perf.
- Speculate the period of data D_b should be long enough to reliably determine whether an item will be consumed in the future.
- Second, the long period of the frequency bins generally shows better classification performances. If the period too short, will makes feature of items noisy.
- Therefore, adjusting these two periods is essential to successfully predicting the interest sustainability of items.

Experiments

Qualitative Results

- Show distribution of features assigned by the neural classifier as the positive or negative class.
- Select the most confident 100 predictions for each class tends to gradually increase over time, positive class tends to decrease over time.
    - Reassert the necessity of the sequential feature and the sequence encoder to capture the temporal dynamics of users' pattern.

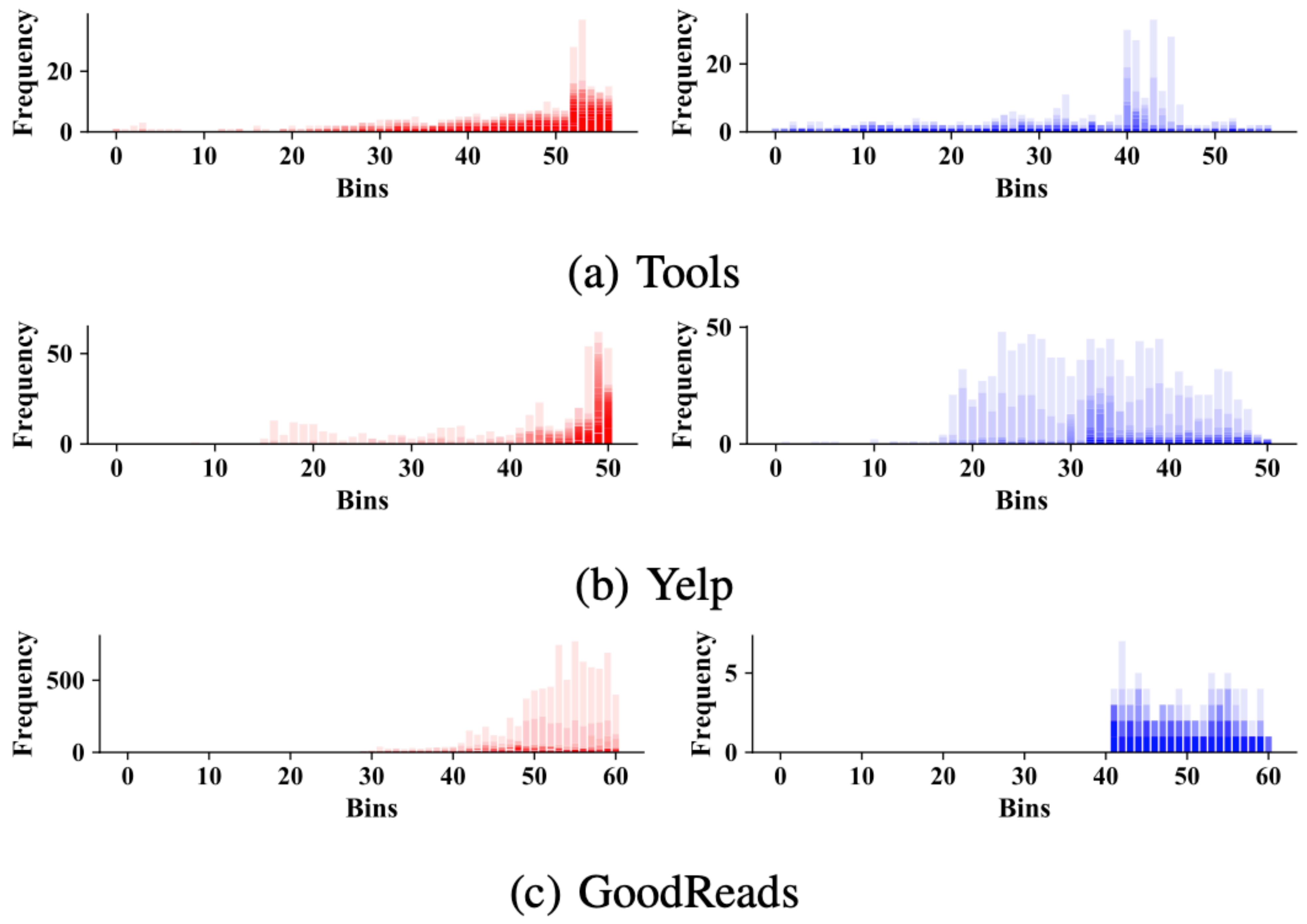


Fig. 9: Distribution of features associated with the most confident 100 predictions in each class. Left and right figures represent the features classified as positive and negative, respectively.

Conclusions

- Build a recommender system based on the ISS (interest sustainability score).
- First predict the interest sustainability of items to obtain the ISS for each item based on a neural classifier.
- Afterward, build a recommender system based on the metric learning framework with the ISSs of items to capture the concept drift of users.
- Reveal that the ISSs are indeed crucial to boost the accuracy of recommendations.

Comments
of Interest Sustainability-Aware Recommender System

- recommend items based on the interest sustainability score (ISS)
- Do many experiments
- Training depend less on noise ISS, but evaluation depend on de-noised ISS.