北京工業大學
BEIJING UNIVERSITY OF TECHNOLOGY

SMU
SINGAPORE MANAGEMENT UNIVERSITY

An aspect performance-aware hypergraph neural network for review-based recommendation

Junrui Liu, Tong Li*, Di Wu, Zifang Tang, Yuan Fang, Zhen Yang

The 18th WSDM 2025

Outline

- Introduction
- Method
- Experiments
- Conclusions & Future work

1. Introduction

1.1 Background

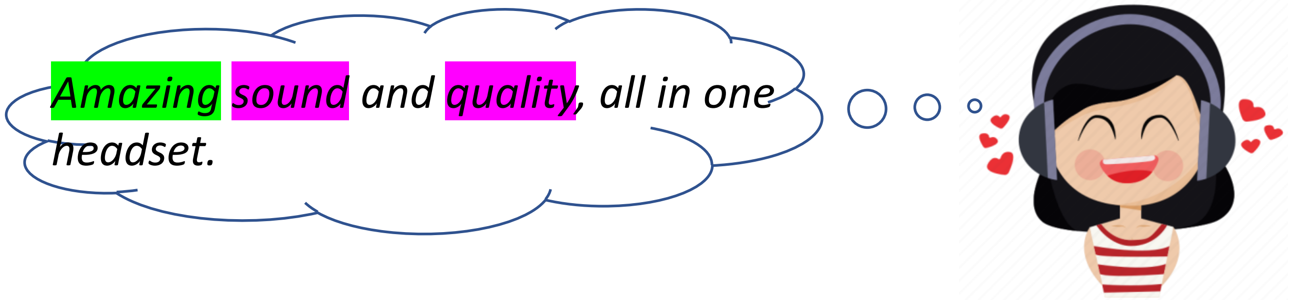
Recommender systems:

- enhance user experiences of web services
- encourage users to share their feelings through ratings and reviews.

Aspect :

- a word or phrase that describes a property of items,
- explicitly describes the characteristics of the items that the user cares about.

Users usually give their opinions or sentiments about special aspects in reviews.



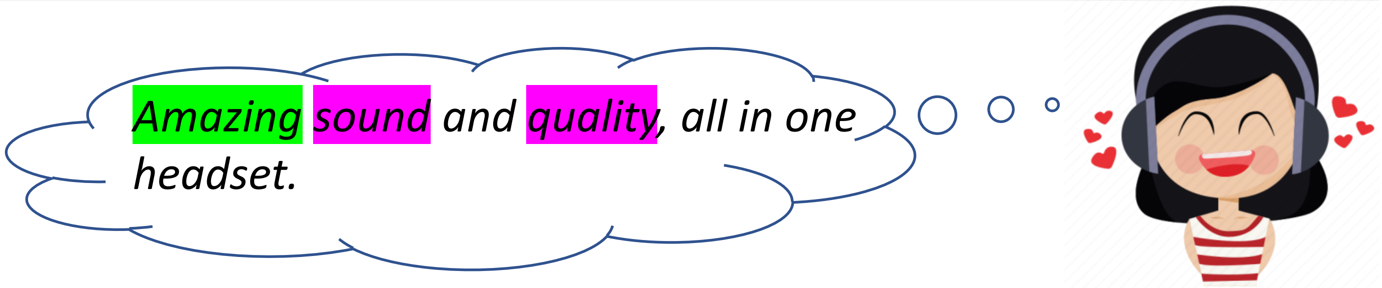
1.2 Progress

- Modeling textual semantics
  - Topic model (CTR, CDL, etc.)
  - CNN (ConvMF, TEART, etc.)
- Modeling aspect-level preference
  - Semantic similarity (APRE)
  - Attention (ANR, MRCP)
  - GNN (MA-GNNs, RGNN)

1.3 Shortcomings

- Modeling users' fine-grained preferences to specific item features
- Identifying the importance of aspects
  - Similarity between the aspects and their content feature
  - equal
  - attention

Users select items, they tend to prioritize performance in various aspects



1.4 Challenge

- Not all the performances can be directly obtained.
- Users' opinions on a specific aspect of an item partially reflect the item's performance in that aspect, but **conflicts in the sentiment polarities expressed by different users**

u_1

This headphone is worth the price, but the tone quality is not good enough, it doesn't play low frequencies very well.

u_2

It is quite expensive, the price is almost 2x than a normal one. Best tone quality I've ever used.

u_3

The size of this headphone is just right, and my ears are not uncomfortable when I wear it all day.

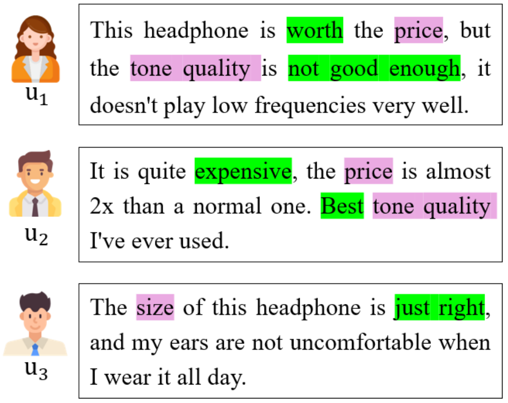
	User	Item	Aspect	Sentiment Polarity
e_1	u_1	i	a_1	Pos
e_2	u_2	i	a_1	Neg
e_3	u_1	i	a_2	Neg
e_4	u_3	i	a_2	Pos
e_5	u_2	i	a_3	Pos

a_1: $price$ a_2: $tone\ quality$ a_3: $size$

1.5 Proposal

- **Conflicting sentiment polarities arise from the users' preferences in both aspects and sentiments**, thus, identifying the true relationship between items and aspects is essential to consider user preferences when aggregating sentiment polarities.

- **Hypergraphs has stronger representation ability of modeling the relationships**

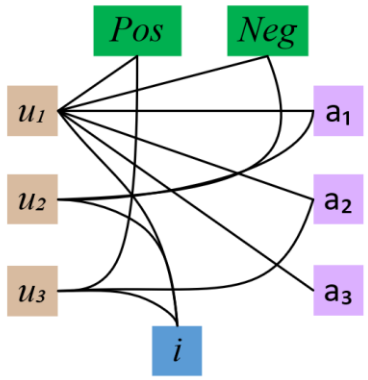


User reviews

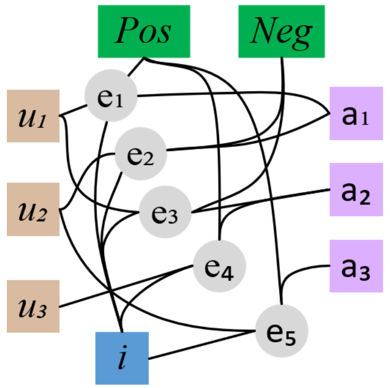
	User	Item	Aspect	Sentiment Polarity
e_1	u_1	i	a_1	*Pos*
e_2	u_2	i	a_1	*Neg*
e_3	u_1	i	a_2	*Neg*
e_4	u_3	i	a_2	*Pos*
e_5	u_2	i	a_3	*Pos*

a_1: *price* a_2: *tone quality* a_3: *size*

Extracted relationships



Simple graph



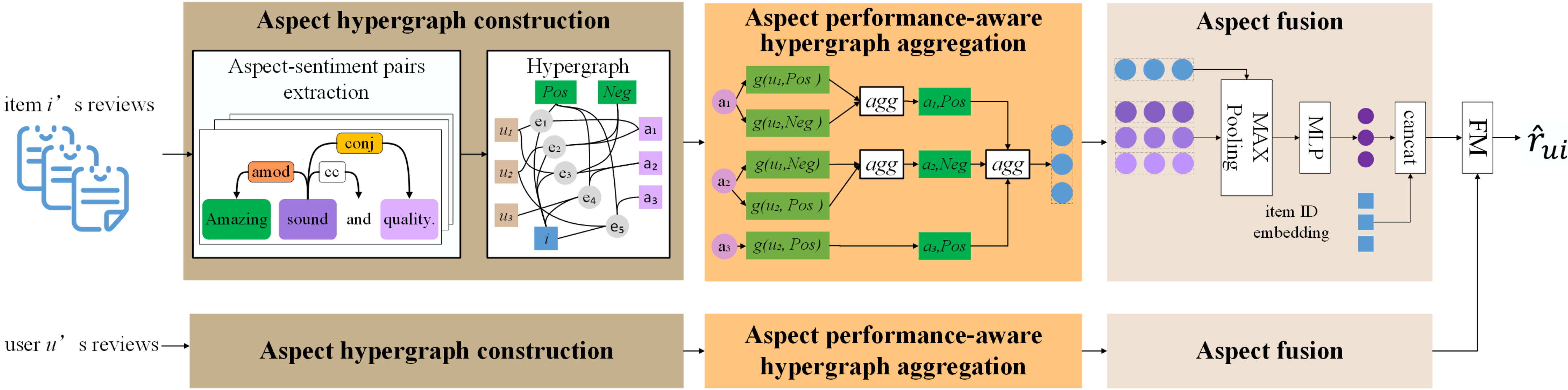
Hypergraph

1.6 Contribution

- An aspect performance-aware hyper graph neural network (APH) for the review-based recommendation, which learns the performance of items from the conflicting sentiment polarity of user reviews.
- An aspect performance-aware hypergraph aggregation method that learns the performances of items on different aspects from conflicting emotional polarities
- Outperforming the best baseline on MSE, Precision@5, and Recall@5 by an average of 2.30%, 4.89%, and 1.60% .

2. Method

2.1 Overview



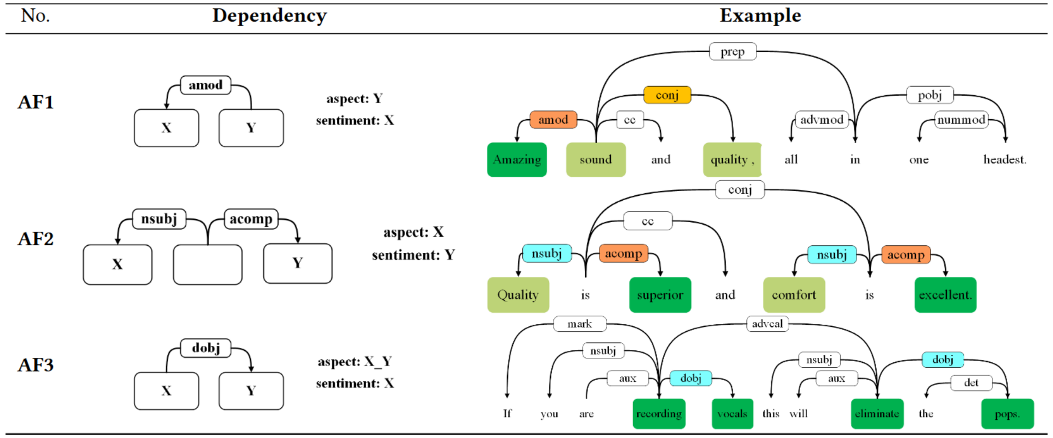
1. Aspect hypergraph construction
2. Aspect performance-aware hypergraph aggregation
3. Aspect fusion
4. FM

2.2 Aspect Hypergraph Construction

- A rule-based unsupervised method: extract aspect sentiment pairs (a, s) from the review of a user u's on an item i.

Table 1: Extracting aspects based on dependency relations

No.	Dependency relations	Aspect	Sentiment
1	Adj. (x) ← amod – Noun (y)	y	x
2	Noun (x) ←nsubj– Linking Verb (y) – – acomp → Adj. (z)	x	z
3	Verb (x) – dobj → Noun (y)	(x,y)	x

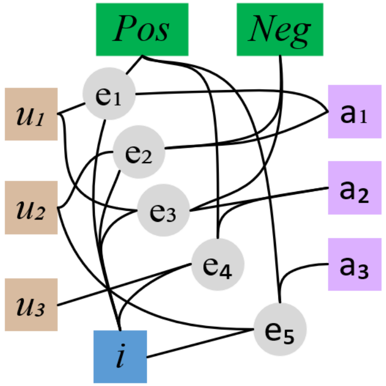


- Construct Hypergraph based on sentiment pairs:

$\mathcal{G} = (\mathcal{V}, \mathcal{E}, \phi)$

ϕ is a node type function

a hyperedge is (u, i, a, s)



Hypergraph vs. simple graph.

u_1: This headphone is worth the price, but the tone quality is not good enough, it doesn't play low frequencies very well.

u_2: It is quite expensive, the price is almost 2x than a normal one. Best tone quality I've ever used.

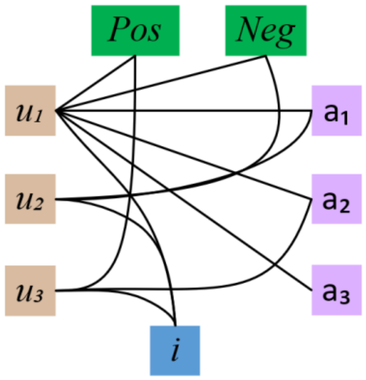
u_3: The size of this headphone is just right, and my ears are not uncomfortable when I wear it all day.

User reviews

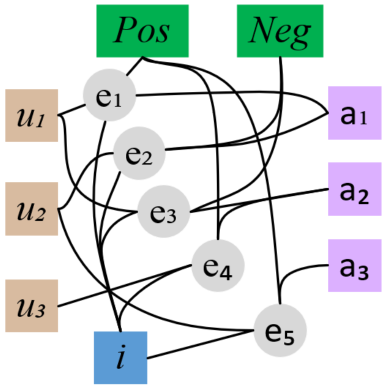
	User	Item	Aspect	Sentiment Polarity
e_1	u_1	i	a_1	*Pos*
e_2	u_2	i	a_1	*Neg*
e_3	u_1	i	a_2	*Neg*
e_4	u_3	i	a_2	*Pos*
e_5	u_2	i	a_3	*Pos*

a_1: *price* a_2: *tone quality* a_3: *size*

Extracted relationships



Simple graph



Hypergraph

Left: The vectorized representation of relationships between users, items, aspects, and sentiment polarities. An relationship set $\mathcal{E} = \{e_1, e_2, e_3, e_4, e_5\}$ and an vertex set $\mathcal{V} = \{u_1, u_2, u_3, i_1, a_1, a_2, a_3, Pos, Neg\}$. The entry (v_i, e_j) is set to 1 if the e_j contains the vertex v_i, and 0 otherwise.

Simple graph: cannot tell us much information, like whether a user has a positive sentiment for what.

Hypergraph: completely illustrates the complex relationships

2.3 Aspect performance-aware Hypergraph aggregation

- The form of existing method to aggregate aspects

$$\mathbf{x}_i = f(\mathcal{A}_i) = \sum_{a \in \mathcal{A}_i} w(\mathbf{x}_a) \cdot \mathbf{x}_a, \quad (1)$$

- Considering the impact of aspect performance

$$\mathbf{x}_i = f(\mathcal{A}_i) = \sum_{a \in \mathcal{A}_i} w(\mathbf{x}_a, q_i(a)) \cdot \mathbf{x}_a, \quad (2)$$

- Learning aspect performance from user preference and their sentiments on aspects

$$\mathbf{x}_i = f(\mathcal{A}_i, g_i(\mathcal{S}_i, \mathcal{U}_i)) = \sum_{a \in \mathcal{A}_i} \sum_{u,s \in \mathcal{U}_i, \mathcal{S}_i} w(\mathbf{x}_a, g_i(\mathbf{x}_u, \mathbf{x}_s)) \cdot \mathbf{x}_a, \quad (3)$$

- The implementation of our aggregation method

$$\hat{\mathbf{x}}_i = \sum_{\mathcal{E}_{i,a} \in \mathcal{E}_i} \sum_{e \in \mathcal{E}_{i,a}} w(e) \mathbf{x}_a \mathbf{W}_4, \quad (7)$$

$$w(e) = w(u, i, a, s) = \frac{exp[\pi(\mathbf{x}_i, g_i(\mathbf{x}_u, \mathbf{x}_s), \mathbf{x}_a)]}{\sum_{e' \in \mathcal{E}_{i,a}} exp[\pi(\mathbf{x}_i, g_i(\mathbf{x}_{u'}, \mathbf{x}_{s'}), \mathbf{x}_a)]}, \quad (4)$$

$$\mathbf{x}_q = q_i(\mathbf{x}_u, \mathbf{x}_s) = MLP(\mathbf{x}_u, \mathbf{x}_s),$$

$$\pi(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3) = LeakyRelu[(\mathbf{x}_1 \mathbf{W}_1)(\mathbf{x}_2 \mathbf{W}_2 + \mathbf{x}_3 \mathbf{W}_3)] \quad (6)$$

2.4 Aspect Fusion

- Users may decide whether to buy an item because of its extreme performance in an aspect or the item's overall performance.

- Learning more about the role of aspects in user-item interactions,

$$\hat{\mathbf{g}}_u = max_{t=1}^{d_2} \hat{\mathbf{X}}_u(:, t), \quad (9)$$

$$\mathbf{g}_u = ReLU(\hat{\mathbf{g}}_u \mathbf{W}_6 + b_6), \quad (10)$$

$$\mathbf{m}_i = ReLU(\hat{\mathbf{x}}_i \mathbf{W}_7 + b_7), \quad (10)$$

$$\mathbf{y}_i = \mathbf{m}_i \oplus \mathbf{g}_i, \quad (11)$$

2.5 Prediction

- User a FM layer to predict with user feature p_u and item feature q_i

$$\mathbf{z} = \mathbf{y}_u \oplus \mathbf{y}_i$$

$$\hat{r}_{ui} = b_0 + b_u + b_i + \mathbf{z}\,\mathbf{w}^T + \sum_{i=1}^{d'} \sum_{j=i+1}^{d'} < \mathbf{v}_i, \mathbf{v}_j > \mathbf{z}_i\,\mathbf{z}_j, \qquad (12)$$

3. Experiments

3.1 Experiment setup

- 6 common datasets

Table 1. The statistics of the experimental datasets.

Dataset	#Users	#Items	#Ratings/Reviews	#Density
Music	1,429	900	10,261	0.80%
Office	4,905	2,420	53,228	0.45%
Toys	19,412	11,924	167,597	0.07%
Beauty	22,363	12,101	198,502	0.07%
Games	24,303	10,672	231,577	0.09%
Yelp	26,084	65,786	3,519,533	0.04%

- 2 types of evaluation metrics
  - MSE、NDCG
  - Precision, Recall

3.2 Baselines

- Rating-based methods:
  - PMF,
  - SVD++
- Review-based methods:
  - CDL,
  - DeepCoNN (DCN) ,
  - NARRE ,
  - CARL ,
  - DAML ,
  - NRCA,
  - DSRLN
- Aspect-based methods:
  - ANR ,
  - MA-GNNs ,
  - RGNN

3.2 Baselines

- PMF [28] is the probabilistic matrix factorization model, which is a classical collaborative filtering-based rating prediction method.
- SVD++ [17] is a classic matrix factorization method that exploits both the user's explicit preferences on items and the influences of the user's historical items on the target item
- CDL [36] is a hierarchical Bayesian model that employs SDAE for learning features from the content information and collaborative filtering for modeling the rating behaviors.
- DeepCoNN (DCN) [42] contains two parallel networks, which focus on modeling the user behaviors and learning the item properties from the review data.
- NARRE [3] uses an attention mechanism to model the importance of reviews and a neural regression model with review-level explanations for rating prediction.
- CARL [38] is a context-aware representation learning model for rating prediction, which uses convolution operation and attention mechanism for review-based feature learning and factorization machine for modeling high-order feature interactions.
- DAML [21] employs CNN with local and mutual attention mechanisms to learn the review features and improve the interpretability of the recommendation model.
- NRCA [23] uses a review encoder to learn the review representation and a user/item encoder with a personalized attention mechanism to learn user/item representations from reviews.
- DSRLN [25] extracts static and dynamic user interests by stacking attention layers that deal with sequence features and attention encoding layers that deal with of user-item interaction.
- ANR [6] is an aspect-based neural recommendation model that learns aspect-based representations for the user and item by an attention-based module. Moreover, the co-attention mechanism is applied to the user and item importance at the aspect level.
- MA-GNNs [41] predefines four aspects and constructs multiple aspect-aware user-item graphs, regarding the aspect-based sentiment as the edge. As it is trained by pairwise loss, we only compared it with NDCG.
- RGNN [26] builds a review graph for each user where nodes are words and edges are word orders. It uses a type-aware graph attention network to summarize graph information and a personalized graph pooling operator to capture important aspects.

3.3 Comparison results

Table 3: The performances of different recommendation methods evaluated by MSE. The best results are in bold faces and the second-best results are underlined. * indicates that the standard deviation of the results of the five times is smaller than 0.001.

Dataset	Music	Office	Toys	Games	Beauty	Yelp
PMF	1.8783	0.9635	1.6091	1.5260	2.7077	1.4217
SVD++	0.7952	0.7213	0.8276	1.2081	1.2129	1.2973
CDL	1.2987	0.8763	1.2479	1.6002	1.7726	1.4042
DCN	0.7909	0.7315	0.8073	1.1234	1.2210	1.2719
NARRE	0.7688	0.7266	0.7912	1.1120	1.1997	1.2675
CARL	0.7632	0.7193	0.8248	1.1308	1.2250	1.3199
DAML	0.7401	0.7164	0.7909	1.1086	1.2175	1.2700
NRCA	0.7658	0.7343	0.8100	1.1259	1.2034	1.2721
DSRLN	0.7538	0.7131	0.8141	1.1205	1.1951	<u>1.1655</u>
ANR	0.7825	0.7237	0.7974	1.1038	1.2021	1.2708
RGNN	<u>0.7319</u>	<u>0.7125</u>	**0.7786**	<u>1.0996</u>	<u>1.1885</u>	1.2645
APH	**0.6795***	**0.6884***	<u>0.7859*</u>	**1.0829**	**1.1757***	**1.1467***

Table 4: The performances of different recommendation methods evaluated by NDCG@10. The best results are in bold faces and the second-best results are underlined. * indicates that the standard deviation of the results of the five times is smaller than 0.001.

Dataset	Music	Office	Toys	Games	Beauty	Yelp
DCN	0.977	0.973	0.975	0.971	0.966	0.941
NARRE	0.978	0.976	0.981	0.968	0.971	0.957
CARL	0.980	0.978	0.978	0.969	0.966	0.943
DAML	<u>0.982</u>	0.978	0.979	**0.979**	0.967	0.958
DSRLN	0.781	0.974	0.977	**0.979**	0.967	0.948
MA-GNNs	0.979	0.973	0.975	0.966	0.965	0.933
RGNN	<u>0.982</u>	<u>0.983</u>	<u>0.982</u>	0.976	<u>0.973</u>	<u>0.963</u>
APH	**0.988***	**0.986***	**0.983***	<u>0.977*</u>	**0.974***	**0.965***

- Achieves the best results compared with other baselines in most case and one second-best
- APH can effectively improve prediction performance by modeling the performance of items in aspects.

3.3 Comparison results

Table 5: The performances of different recommendation methods evaluated by P@5 and R@5. The best results are in bold faces and the second-best results are underlined. * and ‡ indicate that the standard deviation of the results of the five times is smaller than 0.001 and 0.002, respectively.

	Music		Office		Toys		Games		Beauty		Yelp	
	Pre@5	Rec@5	Pre@5	Rec@5	Pre@5	Rec@5	Pre@5	Rec@5	Pre@5	Rec@5	Pre@5	Rec@5
DCN	0.2327	0.6818	0.2555	0.5953	0.2408	0.6228	0.2561	0.6355	0.2876	0.7024	0.3238	0.5985
NARRE	0.2502	0.6603	0.3265	0.7361	0.0105	0.0341	0.2053	0.4984	0.1545	0.4094	0.3976	0.5907
DAML	0.2515	0.7019	0.3158	0.6796	0.2517	0.6638	0.2598	0.6622	0.2227	0.5911	0.3861	0.6138
RGNN	0.2690	0.7453	0.3229	0.6967	0.2874	0.7599	<u>0.2809</u>	<u>0.7164</u>	0.2985	0.7387	0.3824	0.6592
DSRLN	<u>0.2721</u>	<u>0.7518</u>	<u>0.3386</u>	<u>0.7386</u>	<u>0.2873</u>	<u>0.7503</u>	0.2673	0.7131	<u>0.3044</u>	<u>0.7642</u>	<u>0.4278</u>	**0.7248**
APH	**0.2730***	**0.7566‡**	**0.3461***	**0.7433‡**	**0.2985** *	**0.7614***	**0.3263***	**0.7890***	**0.3158***	**0.7753***	**0.4407***	<u>0.6996*</u>

- APH achieves the best results compared with other baselines in six datasets.
- An average improvement 4.89% on Pre@5 and 1.60% on Rec@5.
- For the CTR task, APH can more effectively distinguish the difference between positive and negative items than baseline, by learning the item's performance in certain aspects. The design of APH helps recommender systems recommend more accurately

3.4 Ablation Study

Variants:

- APH(MAX/MEAN) dropouts the aspect performance-aware aggregation layer and uses max/mean pooling instead.
- APH(-AF) dropouts the aspect fusion layer.
- APH(-FM) uses the dot function to predict rating

Table 6: MSE results of ablation study.

Dataset	Music	Office	Toys	Games	Beauty	Yelp
APH(MAX)	0.7006	0.6951	0.7972	1.0879	1.1773	1.1913
APH(MEAN)	0.6933	0.7010	0.7918	1.0799	1.1820	1.1755
APH(-AF)	0.6873	0.7068	0.8040	1.0958	1.1899	1.1869
APH(-FM)	0.8173	0.7196	0.8228	1.1052	1.1999	1.1714
APH	**0.6795**	**0.6884**	**0.7859**	**1.0829**	**1.1757**	**1.1467**

- The aspect performance-aware aggregation accurately learns the performance of items in aspects from user sentiment
- Aspect fusion layer aggregates neighboring aspects of users/items to represent users and items, and learns more about the role of aspects in user-item interactions

3.5 Extracted Aspects

Table 7: The statistics of explicit aspects in various datasets.

Dataset	# Aspect	# Quadruple
Music	601	38,898
Office	3,092	393,038
Toys	4,809	776,819
Games	11,656	2,439,534
Beauty	4,868	866,835
Yelp	43,904	20,857,681

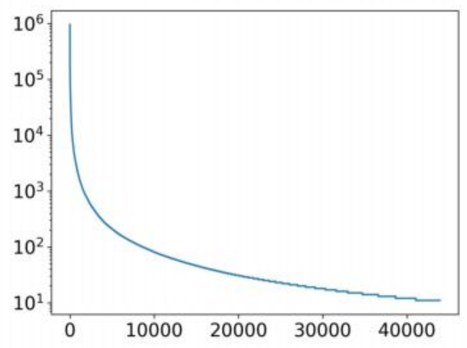


Figure 3: Aspect distribution in the Yelp dataset, which is similar to the other five datasets.

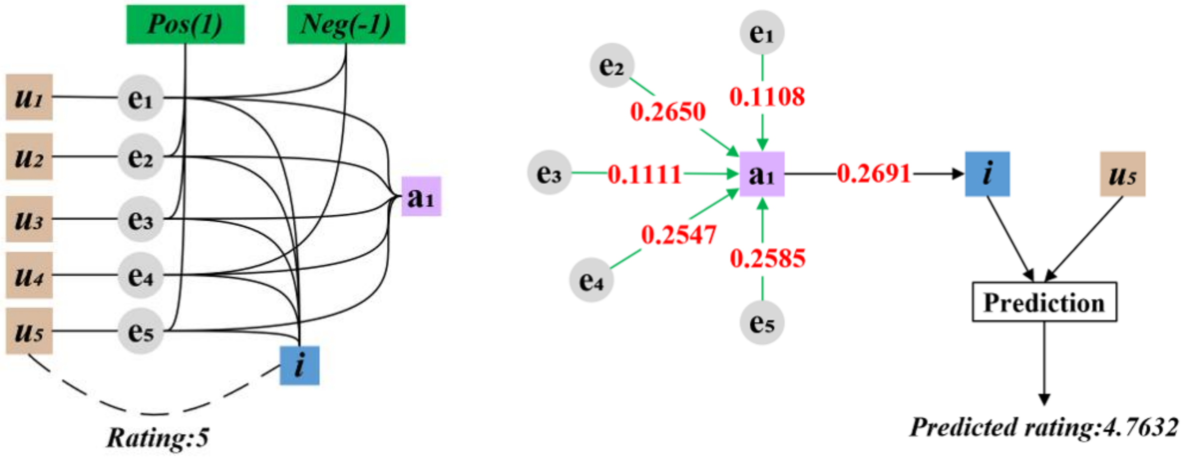
Table 8: Top-10 explicit aspects in various datasets.

Music	Office	Toys	Games	Beauty	Yelp
quality	quality	toy	back	hair	place
guitar	mark	kid	graphic	product	food
draw	color	part	way	scent	service
good	printer	daughter	thing	skin	staff
price	product	boy	quality	color	restaurant
one	price	quality	level	have_hair	selection
wheel	part	back	point	price	price
thing	paper	one	part	have_skin	portion
base	thing	fit	work	face	experience
amp	size	thing	control	smell	sauce

- Effectively extract explicit aspects from reviews.
- The number of aspects is smaller than that of items, and the number of quadruples is bigger than that of ratings.
- Most distributions are long-tail distributions

3.6 Case Study

	Item: *B0000538AC* Aspect: *pack*		
	User	Sent. Polity	Attn. Score
e_1	*A2582KMXLK2P06*	*Neg*	*0.1108*
e_2	*A156P4FPL8OGXB*	*Pos*	*0.2585*
e_3	*A3S15YGZ6W6EV2*	*Pos*	*0.1111*
e_4	*A1S7BFT0HDF3HA*	*Neg*	*0.2547*
e_5	*A3QS4WWC1LCA6H*	*Pos*	*0.2650*



- When the aggregation aspects represent an item, the aspect performance-aware hypergraph aggregation layer calculates the performance of the item in aspects based on the user's sentiment polarities, making the aggregation results more accurate.

3.7 Parameter Sensitivity Study

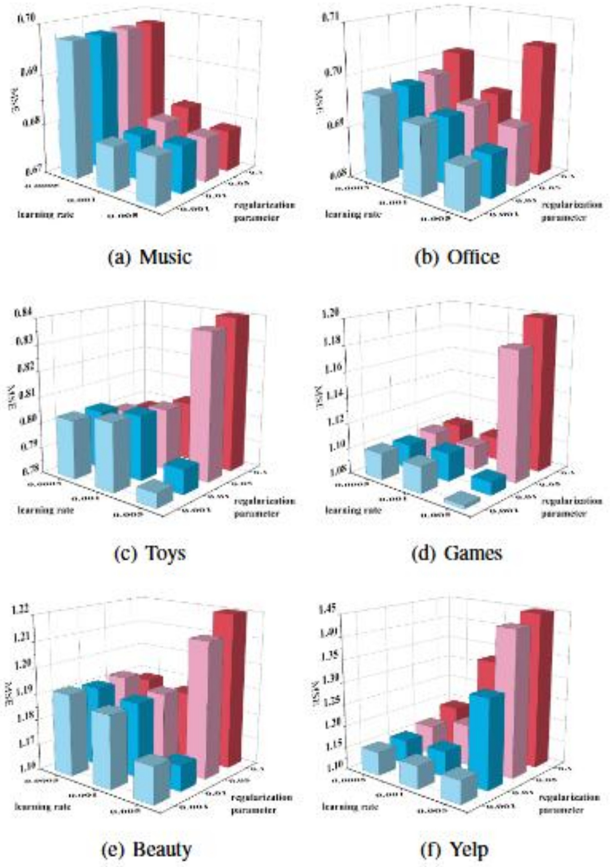


Figure 3. Sparsity analysis of learning rate γ and the regularization parameter λ on six datasets.

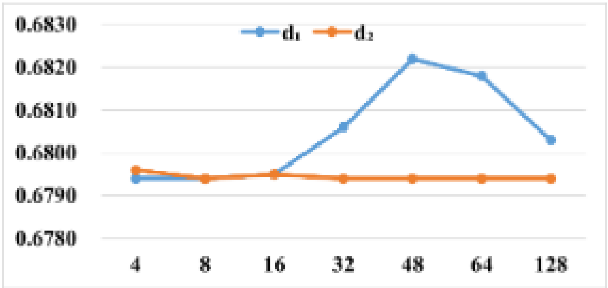


Figure 4. MSE of APM with various dimensions on Music dataset.

- Best value of learning rate is $\gamma = 0.005$ and regularization parameter $\lambda = 0.001$.
- Best dimension are $d_1 = 8$ and $d_2 = 8$

4. Conclusions & Future work

4.1 Conclusions & Future work

- Due to the performances of items on aspects being unavailable in datasets, existing methods **only consider user preferences in aspects reflected in the reviews** when aggregating aspects, and do not consider the actual performance of items in those aspects, leading to suboptimal results.
- We argue that the **performances can be extracted and learned from user reviews**.
- To this end, this paper proposes an aspect performance-aware hypergraph neural network for recommender systems, which considers user preference for aspects and the performance of items in those aspects when calculating their importance. We extract aspect-sentiment pairs from reviews and then construct an aspect-based hypergraph. Subsequently, we design a method that **incorporates user preferences in aspect sentiment pairs to aid in aggregating conflicting sentiment features and learn the item's performance in each aspect**. An aspect fusion layer respectively combines aspects with users and items, **modeling the role that aspects play in the interaction between users and items**.
- Experiments on six real-world datasets demonstrate that the predictions of APH significantly outperform baselines.
- In future work, we plan to extract aspect categories to enhance the connectivity of aspect graphs.

Thanks for listening!