Representing and Recommending Shopping Baskets with Complementarity, Compatibility, and Loyalty

Mengting Wan, Di Wang, Jie Liu, Paul N. Bennett, Julian McAuley
∗ University of California, San Diego † Microsoft Corporation

Oct. 2018

Automated Grocery Shopping

- **Online grocery shopping platforms**
  - Amazon Fresh, Instacart, etc.
- **Physical checkout-free grocery stores**
  - Amazon Go, etc.
- **Mobile scanners, mobile grocery apps**

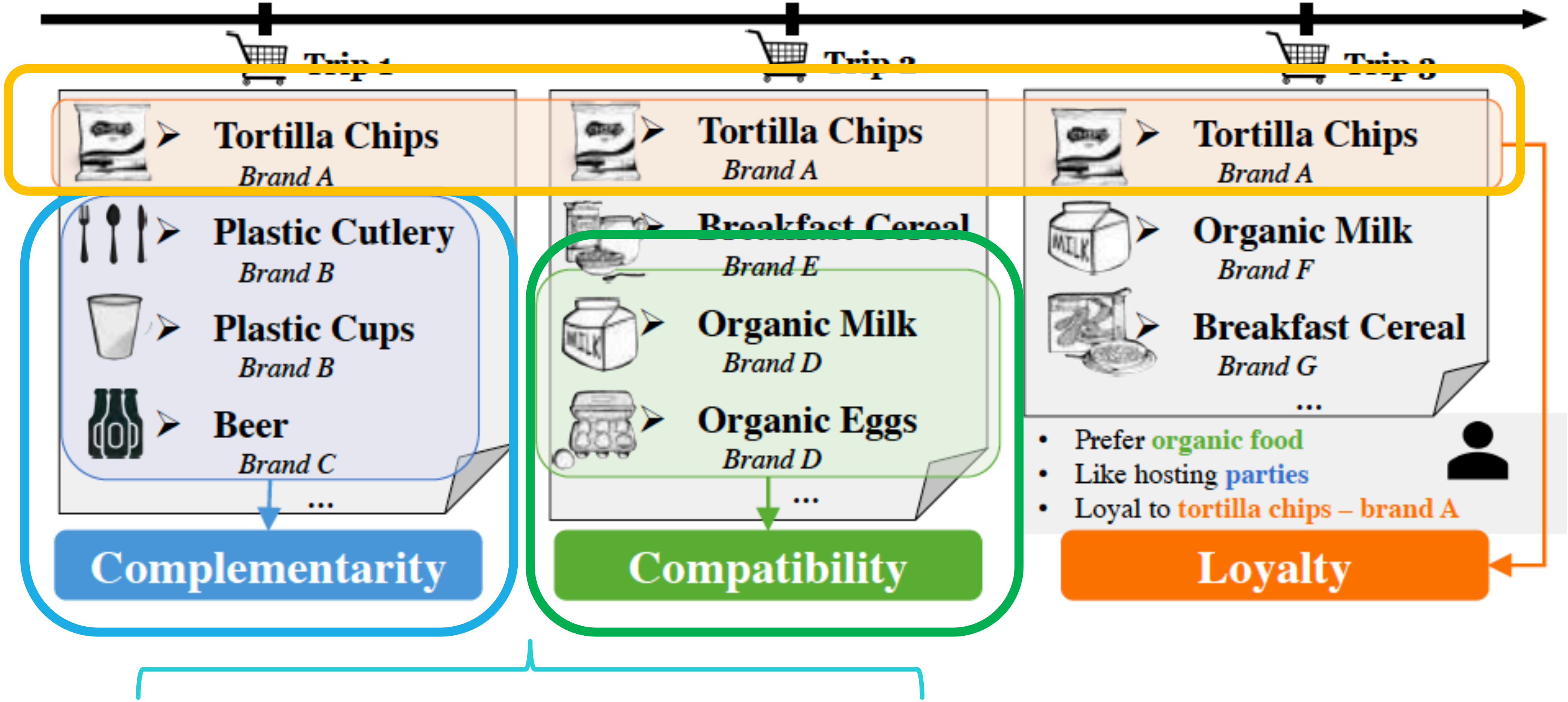
- **Various predictive tasks**
  - Personalized product recommendation
  - Automated product classification
  - Retail sales prediction & inventory optimization
  - etc.

Product Representation & Recommendation

- **Product Representation**
  - Meaningful & Useful
  - **triple2vec**
- **Purchase Prediction / Product Recommendation**
  - Effective & Accurate
  - **adaLoyal**

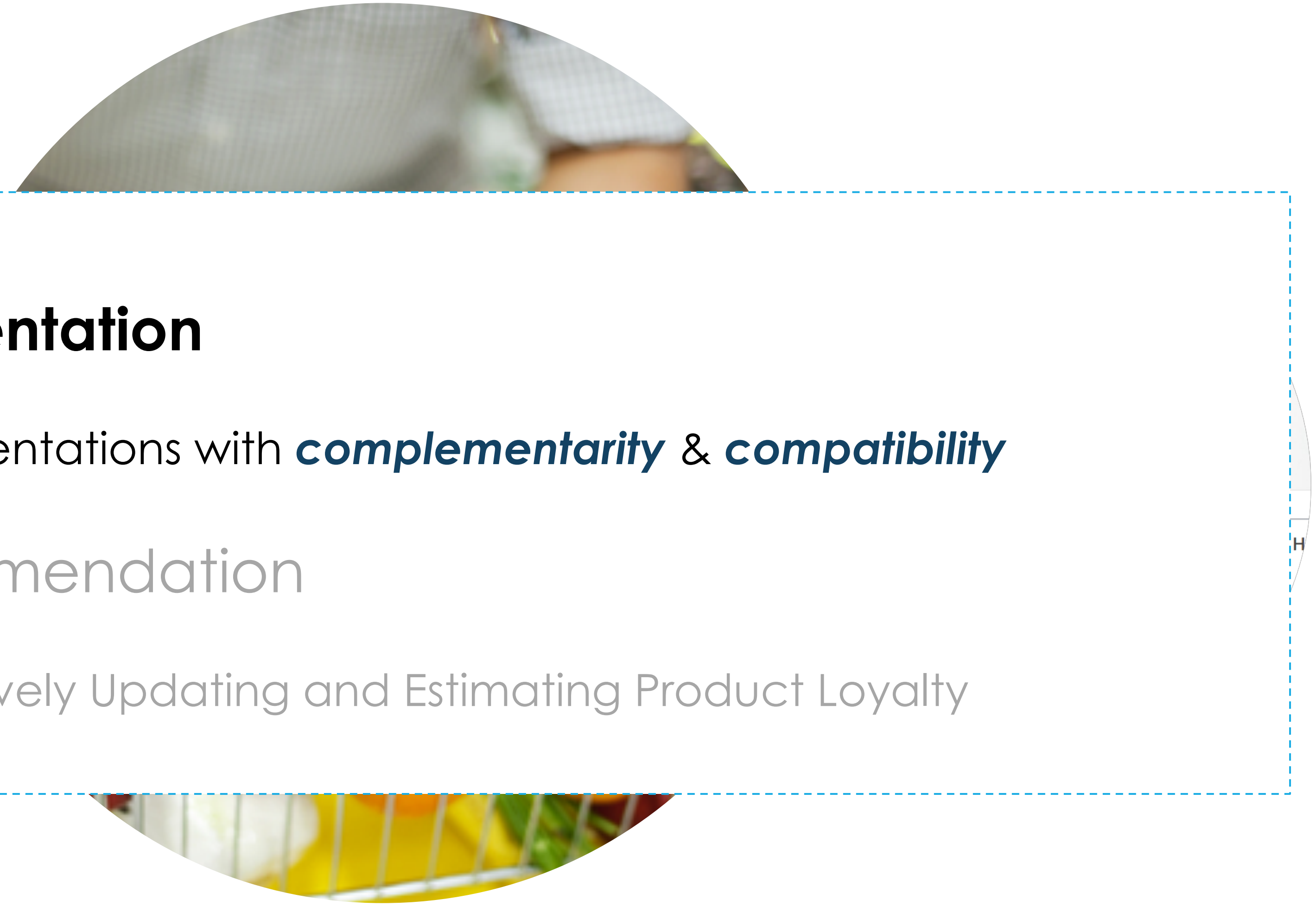
- **Various predictive tasks**
  - Personalized product recommendation
  - Automated product classification
  - Retail sales prediction
  - Retail inventory optimization
  - etc.

Domain-Specific Shopping Patterns



Latent functions for complementarity may need to match users' preferences (user-item compatibility) as well

Grocery Product Representation & Recommendation

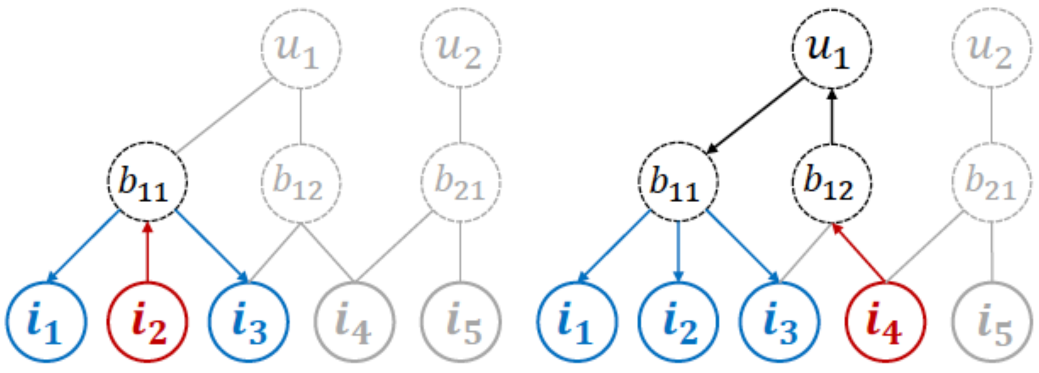


- **Product Representation**
  - triple2vec: representations with ***complementarity*** & ***compatibility***
- Product Recommendation
  - adaLoyal: Adaptively Updating and Estimating Product Loyalty

Product Representation Learning (Background)

- Variants of ***word2vec***
  - Use a target word/product to predict contextual words/products
  - Item2vec, prod2vec, etc.
- Unsupervised Approach
  - Outcome: low-dim representations for each product
- Useful for various downstream tasks
  - Product classification
  - Product recommendation

Context: Same-Basket Context: Same-User



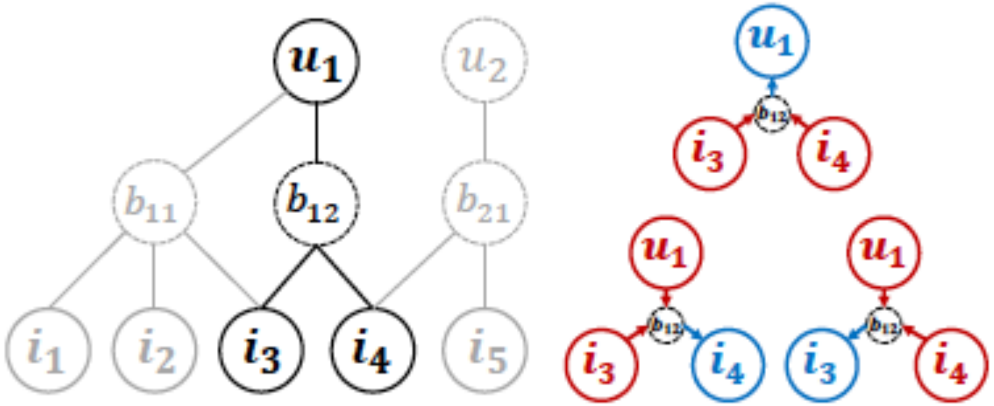
(a) item2vec (b) prod2vec

$$\mathcal{L}_{sgn} = \sum_{v} \sum_{v' \in C_v} \log P(v'|v). \quad P(v'|v) = \frac{\exp(f_v^T g_{v'})}{\sum_{v''} \exp(f_v^T g_{v''})}$$

triple2vec: Representations from Triples

- Cohesion Score: modeling within-basket item-to-item ***complementarity***, and across-basket user-to-item ***compatibility*** together

$$s_{i,j,u} = \overbrace{f_i^T g_j}^{\text{item-to-item complementarity}} + \underbrace{f_i^T h_u + g_j^T h_u}_{\text{user-to-item compatibility}}, \quad P(i|j,u) = \frac{\exp(s_{i,j,u})}{\sum_{i'} \exp(s_{i',j,u})}$$

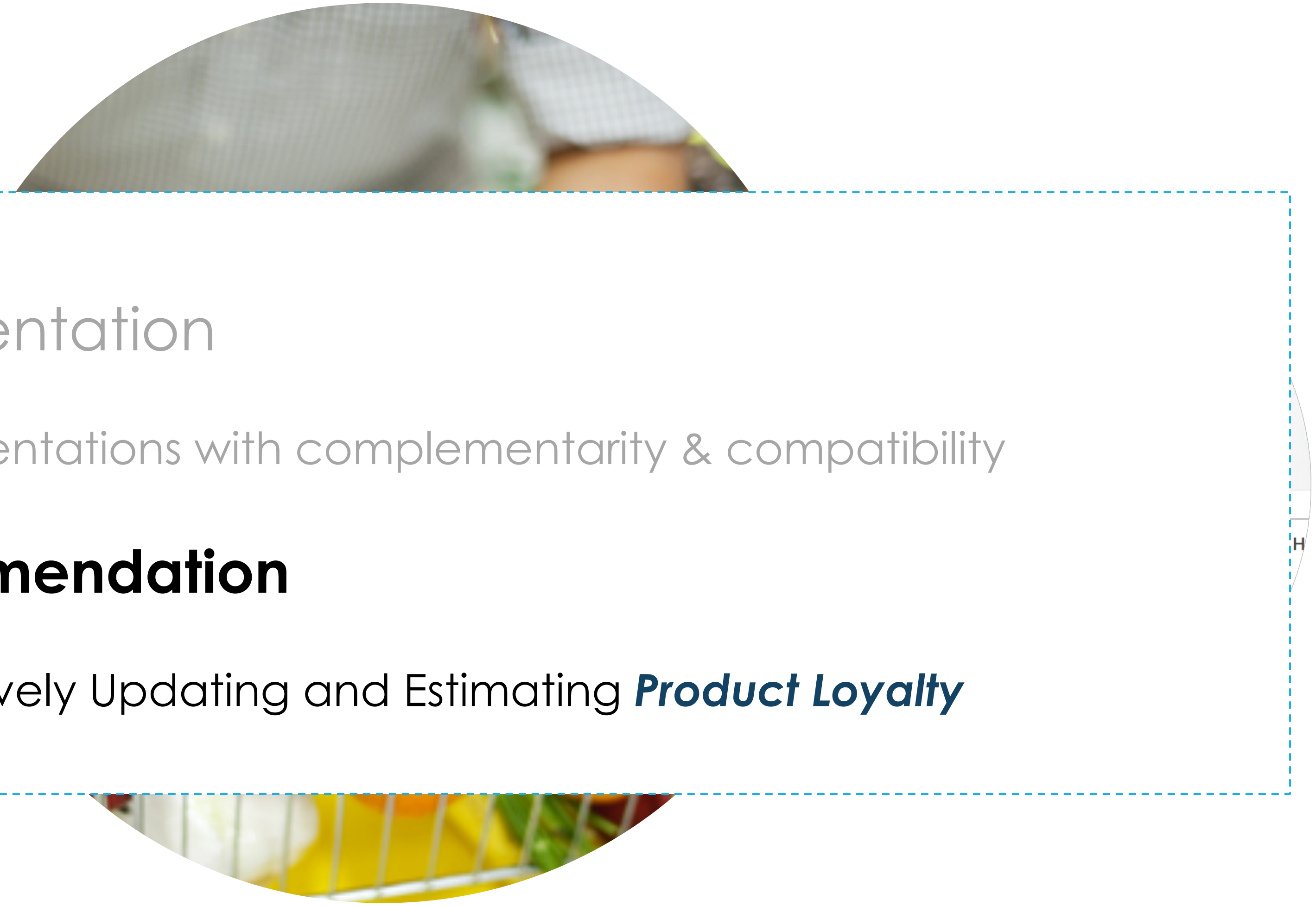


triple2vec

- Iteratively 'knock out' a node and use the other two nodes to predict it

$$\mathcal{L} = \sum_{(i,j,u)\in\mathcal{T}} \big(\log P(i|j,u) + \log P(j|i,u) + \log P(u|i,j)\big),$$

Grocery Product Representation & Recommendation



- Product Representation
  - triple2vec: representations with complementarity & compatibility
- **Product Recommendation**
  - adaLoyal: Adaptively Updating and Estimating ***Product Loyalty***

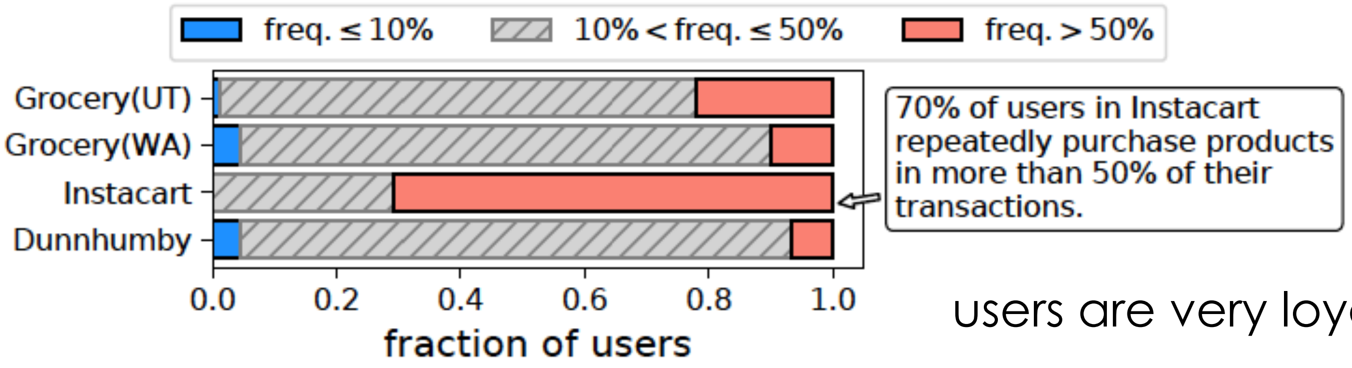
From Product Representation to Recommendation

- Generalize from contextual products to a new product based on their low-dim representations
  - cosine(contextual products, candidate product)
  - inner-product(user, candidate product)
  - cohesion-score(user, contextual products, candidate product)

$$s_{i,j,u} = \overbrace{f_i^T g_j}^{\text{item-to-item complementarity}} + \underbrace{f_i^T h_u + g_j^T h_u}_{\text{user-to-item compatibility}},$$

Generalization & Memorization

- Grocery shopping
  - Numerous re-purchased products
- User-item interaction matrix is not always low-rank
- Significant 'high-rank' patterns can be memorized by counting some statistics (purchase frequency)
- How to balance 'memorization' and 'generalization'?



users are very loyal to their favorite products

Product Loyalty: Preferences Beyond Expectations

- adaLoyal: adaptively estimating product loyalty $\boldsymbol{l_{i,u}^{(t)}}$

An incremental module, can be applied on top of almost any recommenders!

  - Scan a user's transaction logs chronologically:
    - if a *new product* is observed, we activate its corresponding loyalty $\boldsymbol{l_{i,u}^{(t)}}$ and set it to be a given initial value $\boldsymbol{l_0}$
    - if a *product has been purchased before*, $\boldsymbol{l_{i,u}^{(t)}}$ is <u>*updated based on the posterior distribution*</u> of the loyalty indicator
  - Final prediction is a mixture of frequency model and representation model

$$\tilde{p}_{i,u}^{(t)} = l_{i,u}^{(t-1)} \underbrace{\boxed{\mu_{i,u}^{(t-1)}}}_{\text{memorization}} + (1 - l_{i,u}^{(t-1)}) \underbrace{\boxed{p_{i,u}}}_{\text{generalization}}$$

if $x_{i,u}^{(t)} = 1$ **then**

assign $l_{i,u}^{(t)} = \frac{l_{i,u}^{(t-1)}\mu_{i,u}^{(t-1)}}{l_{i,u}^{(t-1)}\mu_{i,u}^{(t-1)} + (1-l_{i,u}^{(t-1)})p_{i,u}}$

else

assign $l_{i,u}^{(t)} = \frac{l_{i,u}^{(t-1)}(1-\mu_{i,u}^{(t-1)})}{l_{i,u}^{(t-1)}(1-\mu_{i,u}^{(t-1)}) + (1-l_{i,u}^{(t-1)})(1-p_{i,u})}$

Experiments (Datasets)

- Dunnhumby
  - public dataset, physical grocery store chain, household-level, frequent shoppers
- Instacart
  - public dataset, online shopping (same-day grocery delivery web service), frequent shoppers
- MSR-Grocery (WA)
  - proprietary dataset, physical convenient store, less frequent shoppers, smaller basket size
- MSR-Grocery (UT)
  - proprietary dataset, physical mid-size stores, mixture of college students and regular households

Experiments
(Product Classification)

Method	*Dunnhumby* micro	*Dunnhumby* macro	*Instacart* micro	*Instacart* macro	*Grocery(WA)* micro	*Grocery(WA)* macro	*Grocery(UT)* micro	*Grocery(UT)* macro
item2vec	0.665	0.108	0.377	0.283	*0.608*	0.345	0.620	0.239
prod2vec	0.617	0.066	0.330	0.218	0.480	0.212	0.491	0.093
m.2vec	0.627	0.071	0.331	0.221	0.441	0.144	0.484	0.067
triple2vec	*0.669*	*0.114*	*0.382*	*0.294*	0.581	*0.361*	*0.623*	*0.293*

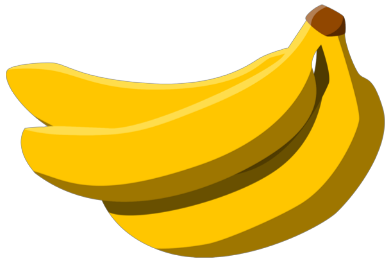
(a) F1 metrics on coarse-grained (department) classification

Method	*Dunnhumby* micro	*Dunnhumby* macro	*Instacart* micro	*Instacart* macro	*Grocery(WA)* micro	*Grocery(WA)* macro	*Grocery(UT)* micro	*Grocery(UT)* macro
item2vec	0.160	0.046	0.187	0.075	0.518	*0.010*	0.275	0.094
prod2vec	0.087	0.015	0.106	0.030	0.518	0.009	0.119	0.023
m.2vec	0.078	0.007	0.155	0.036	0.518	0.007	0.091	0.008
triple2vec	*0.175*	*0.049*	*0.189*	*0.082*	*0.519*	*0.010*	*0.291*	*0.097*

(b) F1 metrics on fine-grained (category) classification

Case Studies (Similarity Search)

Best selling product in a typical grocery store (US)?



the inner product $\boldsymbol{f}_v^T \boldsymbol{g}_v$

cosine similarity based on $\boldsymbol{f}_v + \boldsymbol{g}_v$

Non-Organic

Product Query: "Banana"

Complements	Score	Competitors	Score
Whole Milk With Vitamin D	3.46	Fuji Apple	0.97
Plain Yogurt	3.11	Honeycrisp Apple	0.96
Apple Blueberry Granola	3.06	Cucumber Kirby	0.93
Orange Navel	3.01	Large Lemon	0.92
Milk Chocolate Nutrition Shake	2.99	Large Grapefruit	0.92

Organic

Product Query: "Organic Banana"

Complements	Score	Competitors	Score
Organic Papaya	3.72	Organic Strawberries	0.96
Organic 2% Milk	3.69	Organic Raspberries	0.94
Carbonated Water	3.66	Organic Blueberries	0.94
Organic Bosc Pears	3.61	Organic Hass Avocado	0.93
Organic Applesauce	3.55	Organic Large Extra Fuji Apple	0.92

Complement and competitor search for "Banana" and "Organic Banana" in an online grocery shopping dataset (*Instacart*).

Experiments (Product Recommendation)

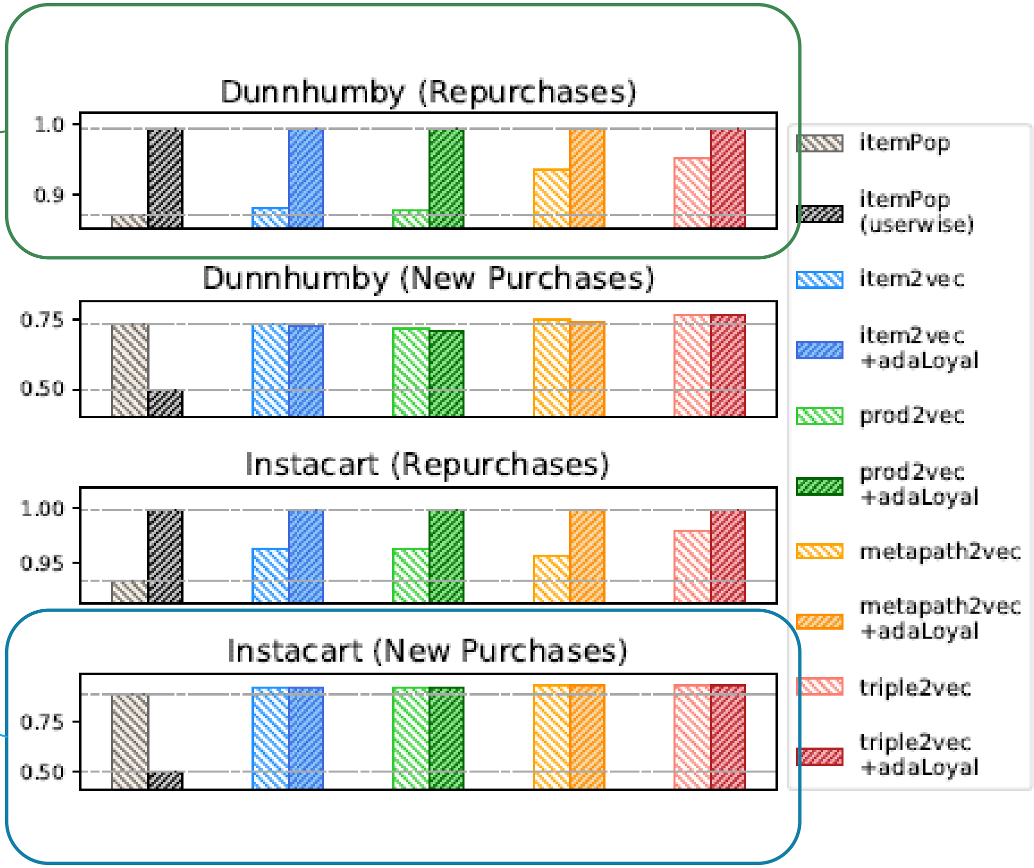
- Two recommendation tasks
  - Next-Basket Recommendation
  - Within-Basket Recommendation
- Evaluation metrics:
  - AUC (overall ranking)
  - NDCG (top-biased ranking)
- Proposed method
  - triple2vec
  - triple2vec + adaLoyal
- Baselines
  - Unsupervised baselines
    - itemPop, user-wise itemPop
    - item2vec, prod2vec, metapath2vec
  - Unsupervised baselines +adaLoyal, +BPR
  - Supervised baselines: BPR-MF, FPMC

Experiments (Product Recommendation)

- triple2vec + adaLoyal generally outperforms other methods
  - Leveraging complementarity, compatibility and loyalty are useful
- Applying adaLoyal consistently & significantly boost recommendation performance
  - Modeling user-product loyalty is very important for grocery shopping recommendation
- Results from user-wise purchase frequency are even better than supervised methods (BPR-MF & FPMC) in terms of NDCG
  - Please do not neglect the naïve method - purchase frequency. Its memorization power is surprisingly useful for recommending grocery products.

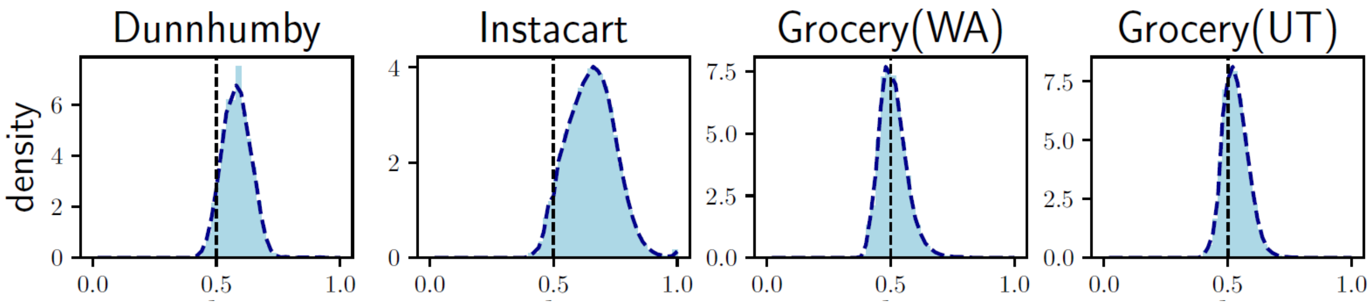
Repurchases vs. New purchases

- How did adaLoyal balance these two?
  - **Repurchases:** boost to the upper bound provided by user-wise itemPop (memorization)
  - **New purchases:** scarifies very limited performance when applying adaLoyal (generalization)



Case Studies (Product Loyalty - User)

User A (l_u = 1.00)	User B (l_u = 0.57)	User C (l_u = 0.37)
Sparkling Water, Bottles	Spinach Artichoke Dip, Taboule Salad, ...	Olive Oil Soap, Citrus Castile Soap, Peppermint Castile Soap ...
Sparkling Water, Bottles	Packaged Grape Tomatoes	Coconut Chips – Sea Salt, Coconut Chips – Original ...
Sparkling Water, Bottles	Bag of Organic Bananas, Taboule Salad	Compostable Forks
Sparkling Water, Bottles	Fuji Apples, Seedless Cucumbers, ...	Grunge Buster Grout And Tile Brush
Sparkling Water, Bottles	Bag of Organic Bananas, Sweet Kale Salad Mix	Pumpkin Seed Cheddar Crispbreads, Seedlander Crispbreads
Sparkling Water, Bottles	Spinach Artichoke Dip, Seedless Red Grapes, ...	Zinc Target Mins 50 Mg Gluten Free Tablets



Conclusions

- Three patterns in users' grocery baskets
  - Complementarity, Compatibility, Loyalty
- A product representation learning method
  - triple2vec
- A recommendation algorithm
  - adaLoyal
- Quantitative and qualitative results product classification& recommendation tasks
- A lot of insights in grocery shopping domain

Thanks!

Mengting Wan (m5wan@ucsd.edu)