Generating Realistic Sequences of Customer-level Transactions for Retails Datasets

Thang Doan
Desautels Faculty of Management
McGill University
Montreal, Canada
thang.doan@mail.mcgill.ca

Neil Veira
Department of Electrical and Computer Engineering
University of Toronto
Toronto, Canada
nveira@eecg.toronto.edu

Brian Keng
Rubikloud Technologies Inc.
Toronto, Canada
brian.keng@rubikloud.com

Outline

- Introduction and Motivation
- Background
- Methodology
- Experiments
- Conclusion

Introduction

- Retailers wish to improve service to increase customer loyalty and sales
- Collect and analyze customer-level transactional data
- Goals:
  - Understand customer's behavior
  - Recommendation systems
  - Targeted customer loyalty programs
  - Relevant advertisement to keep customer subscriptions
  - Offer better discounts

Introduction

Challenges

- Large number of customers and products
- Most customers are involved with multiple retailers
  - Data for a single retailer is censored
- Transactions are sequential in nature

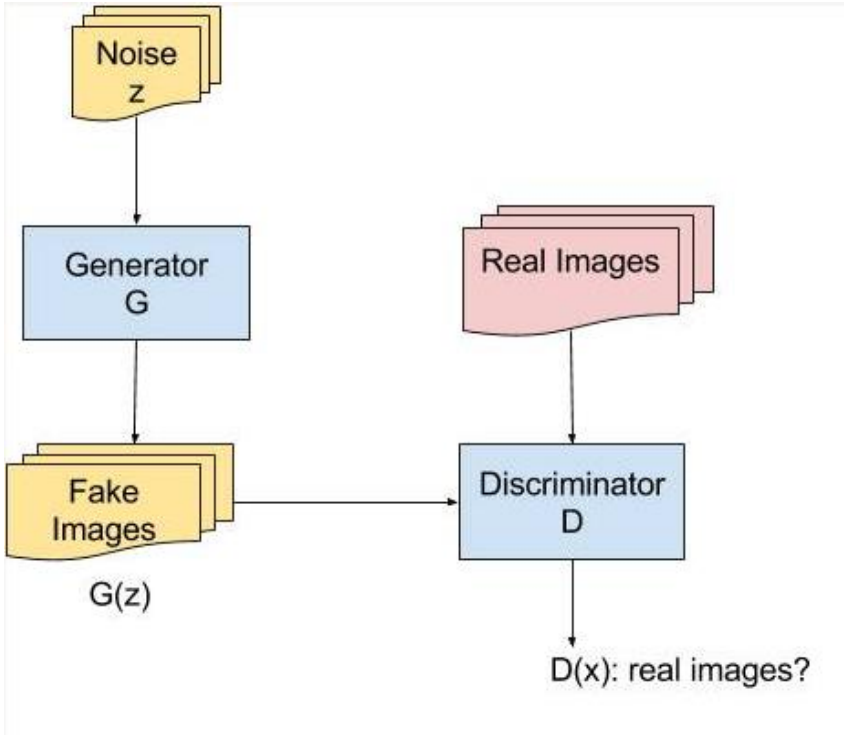
Introduction

- Models that simulate customer behaviour can offer new insights
- Generating transactions directly shows interactions between customers and products
- Transaction sequences consider time-sensitive customer state
  - Future purchases are conditioned on previous ones
- Useful in finding optimal marketing policy

Background: Generative Adversarial Networks (GAN)

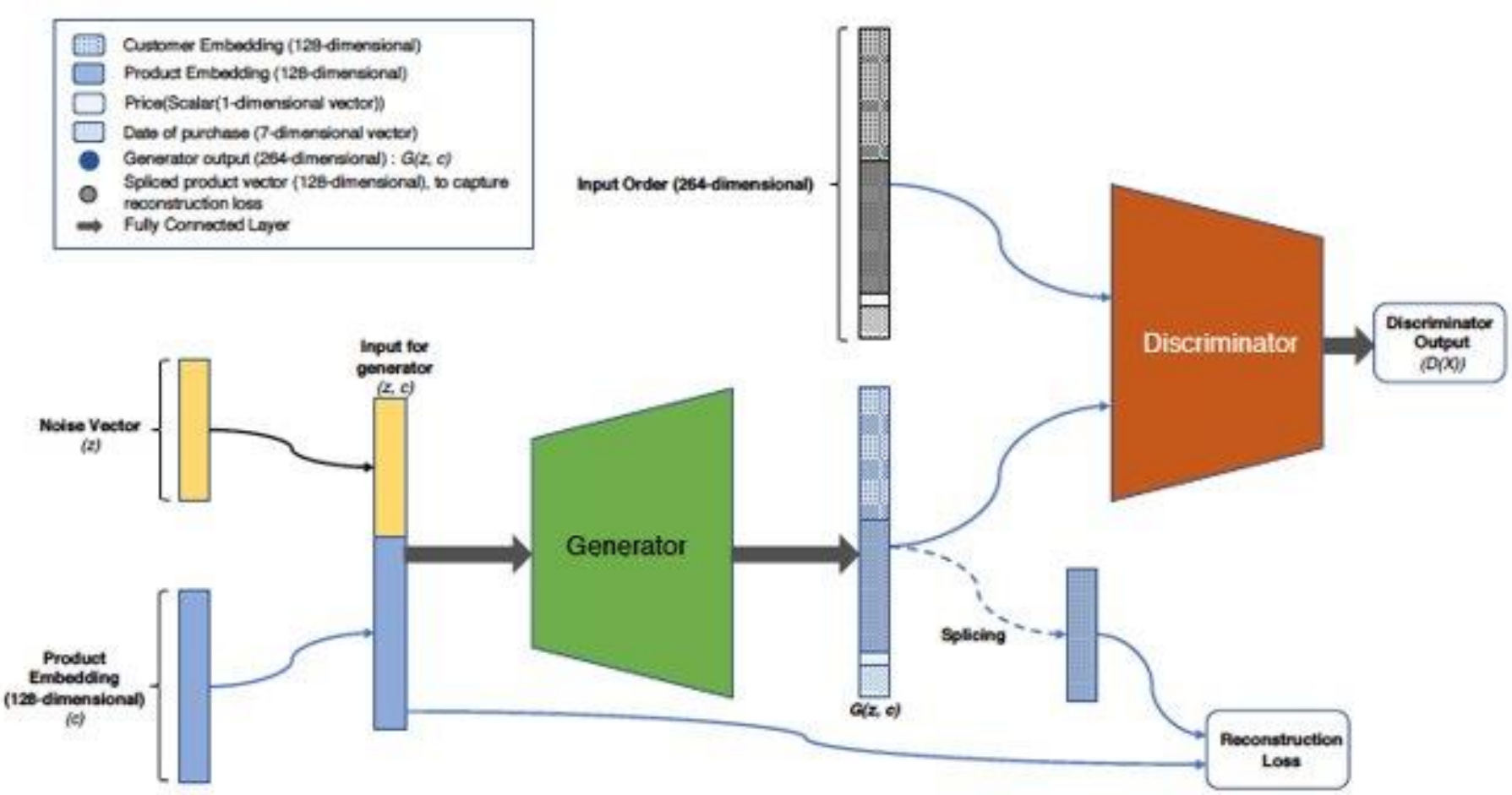
Recall GAN's intuition:

- Goal: learn an underyling distribution
- Trained via 2-player game
- Generator (G): Generate fake samples given input noise
- Discriminator (D): determine if a given sample comes from real or fake distribution
- This dynamic game helps generator produce more realistic samples



Background: ECommerceGAN

- ECommerceGAN: A dynamic recurrent model for next purchase recommendation

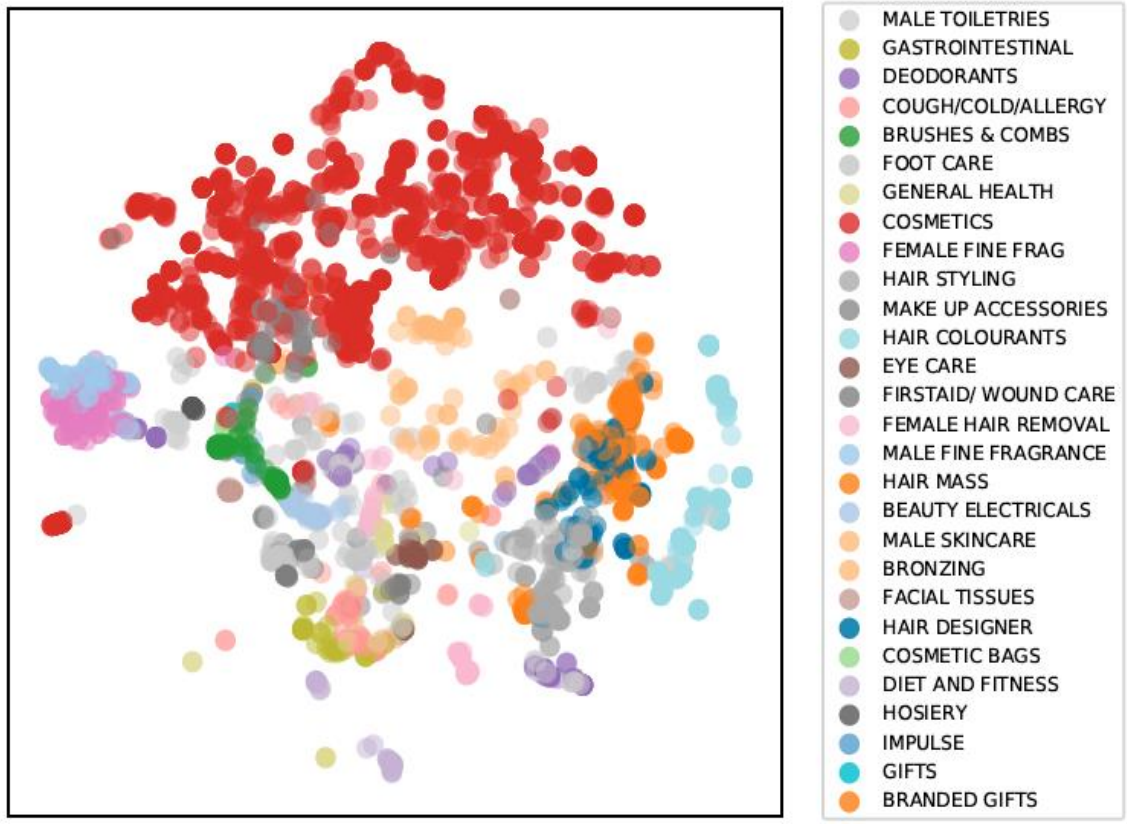


Background: ECommerceGAN

- ECommerceGAN: A dynamic recurrent model for next purchase recommendation
- Given a product, use GAN to generate realistic transaction containing
  - Customer embedding
  - Price
  - Date of purchase
- Can provide insights into product demand
- Only generates individual orders
- Does not directly model customer behaviour over time

Methodology
1) Learning Item Embeddings

- Word2vec: neural embedding model trained on textual descriptions
- Image: Project into 2D space with t-SNE
- Items are classified into functional categories
- Items from same categories are closer



Methodology
2) Learning customer embeddings

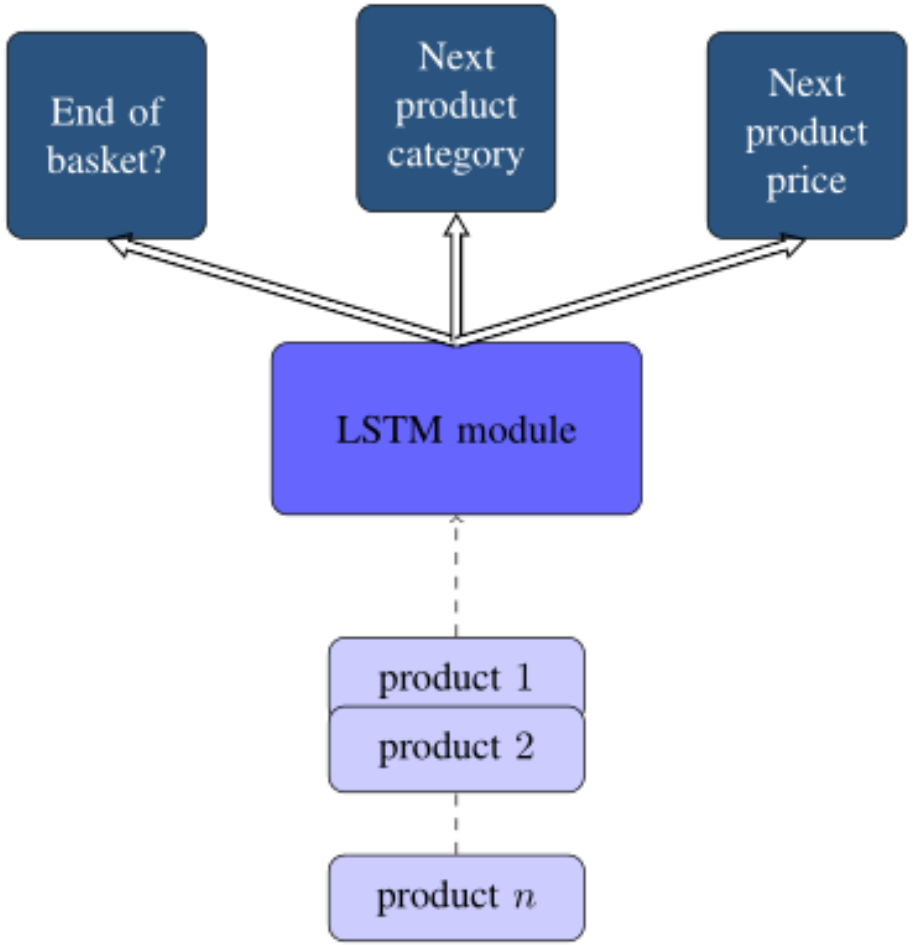
Multi-task optimization

- Feed into LSTM customer's transaction sequence

3 tasks to optimize:

- Predicting end of basket
- Predicting next product category
- Predicting next product price

- Customer embedding is LSTM hidden state
  - Evolves as purchases are made (time-sensitive)



Methodology
3) Training a Conditional WGAN-GP

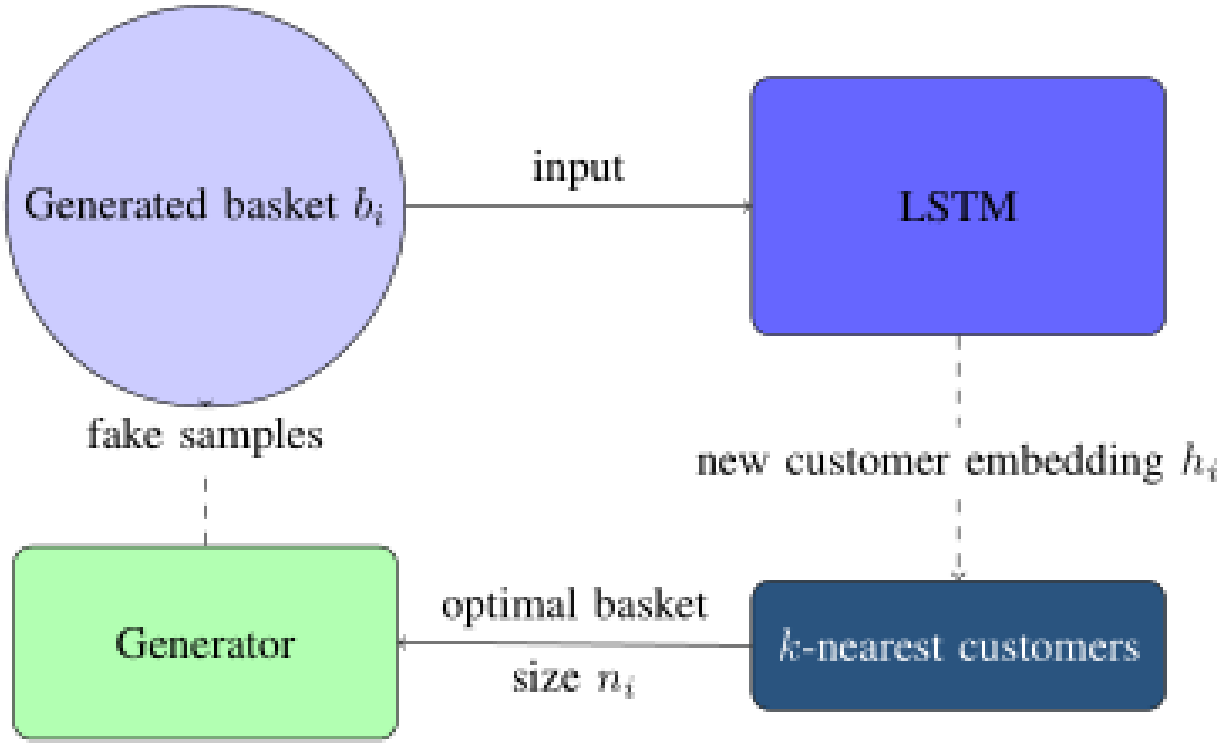
- Conditional on:
  - A customer embedding previously learned h
  - A week number w
- The generator outputs an item embedding
  - Represents next purchase (week w+1) by the customer
- Given an item coordinates the discriminator needs to tell if it is a real transaction for this customer or a fake
- Train GAN to produce realistic next item predictions

Methodology
4) Generating Baskets

- Given: at least one week of transactions for a customer
- Feed into the LSTM, get customer's embedding h
- Find k-nearest customers (L2 distance can be used)
- Compute average basket size among those neighbors, N
  - Prediction of number of items in customer's next basket
- Generate N item embeddings with the GAN
- Map embeddings to actual items in product catalogue
  - Find nearest item to predicted embedding in embedding space

Methodology
5) Pipeline to Generate Sequences

- Feed generated basket for week w+1 into LSTM
- LSTM updates hidden state to customer embedding for week w+1
- Use this to generate basket for week w+2
- Repeat the process
- Result: each week's basket is sensitive to previous weeks' baskets

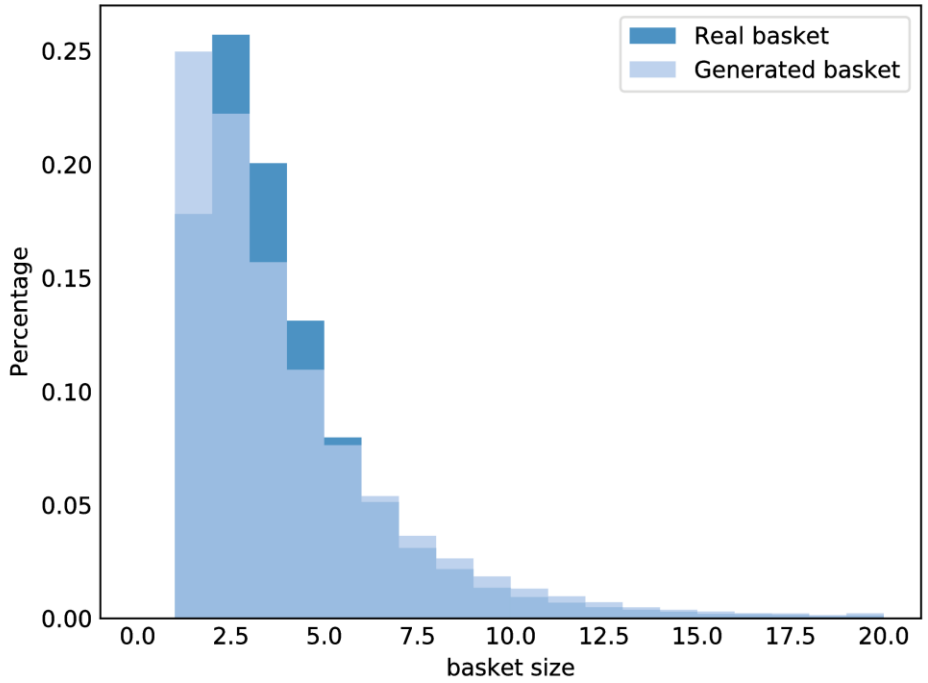


Experimental Results: Dataset

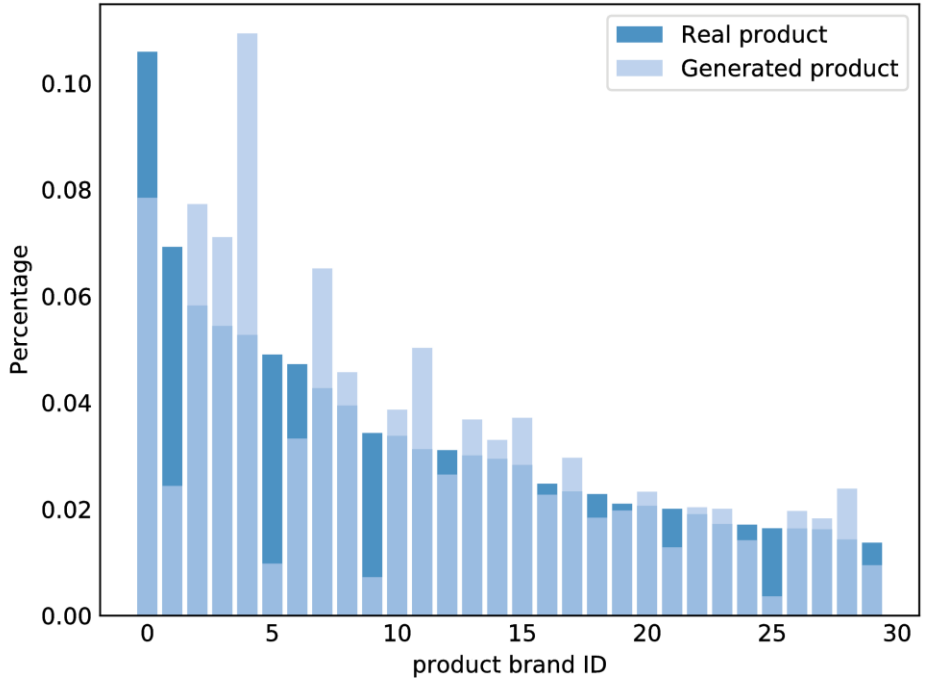
- Transactional data from an industry partner
- Covers 5 weeks during summer of 2016
- 174,301 customer baskets
- 7,722 distinct products
- 66,000 distinct customers
- Average basket size of 4.08 items
- Average price of $12.2

Experimental Results:
Real vs Generated Baskets

Distribution of basket sizes

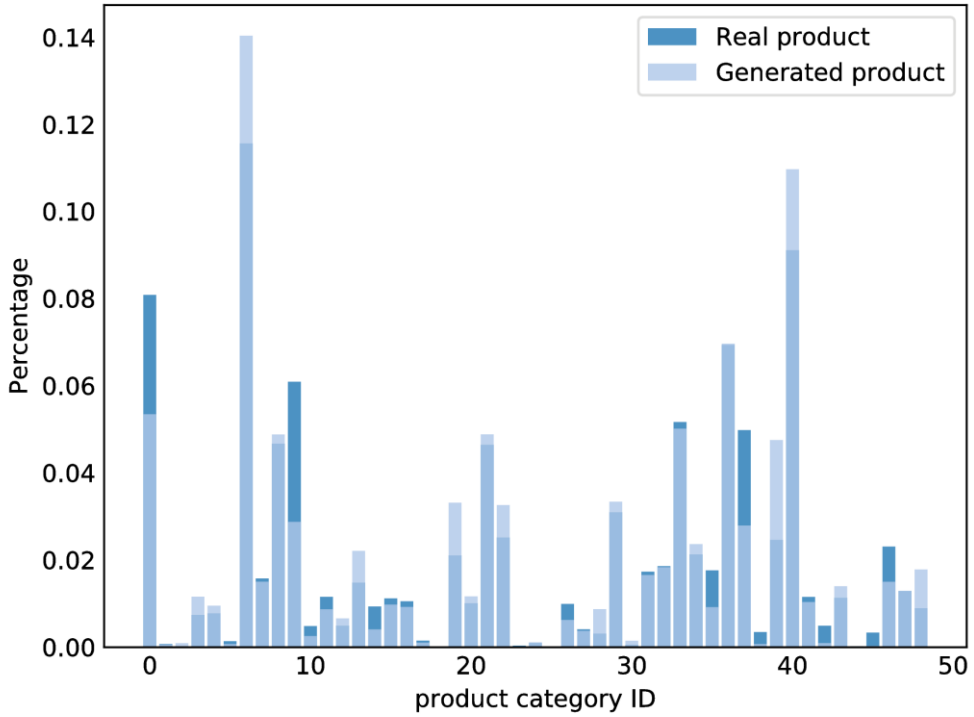


Distribution of brands

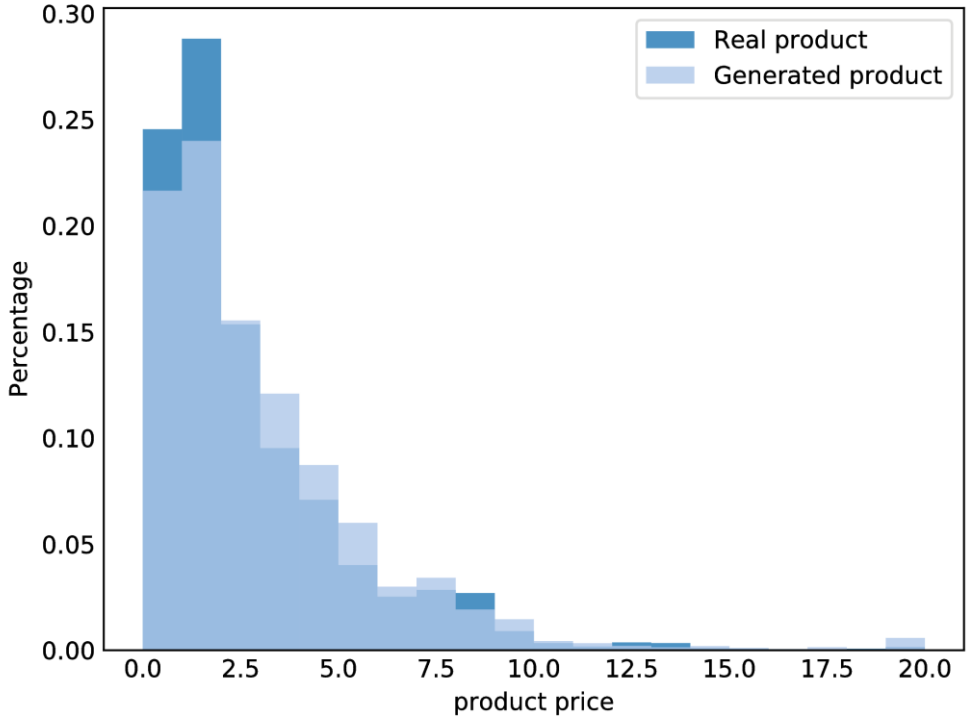


Experimental Results:
Real vs Generated Baskets

Distribution of product types



Distribution of prices



Experimental Results: Basket Statistics

- Statistics from real and generated transactions

	Real Transactions	Generated Transactions
Average basket size	4.08	3.85
Average basket price	$3.1	$3.4

- Max absolute deviation between real and generated baskets

Criterion	Max absolute deviation (in %)
Category	3.2%
Brand	5.6%
Price	5.2%
Basket size*	4.1%

* this metric only applies for basket size ≤ 20

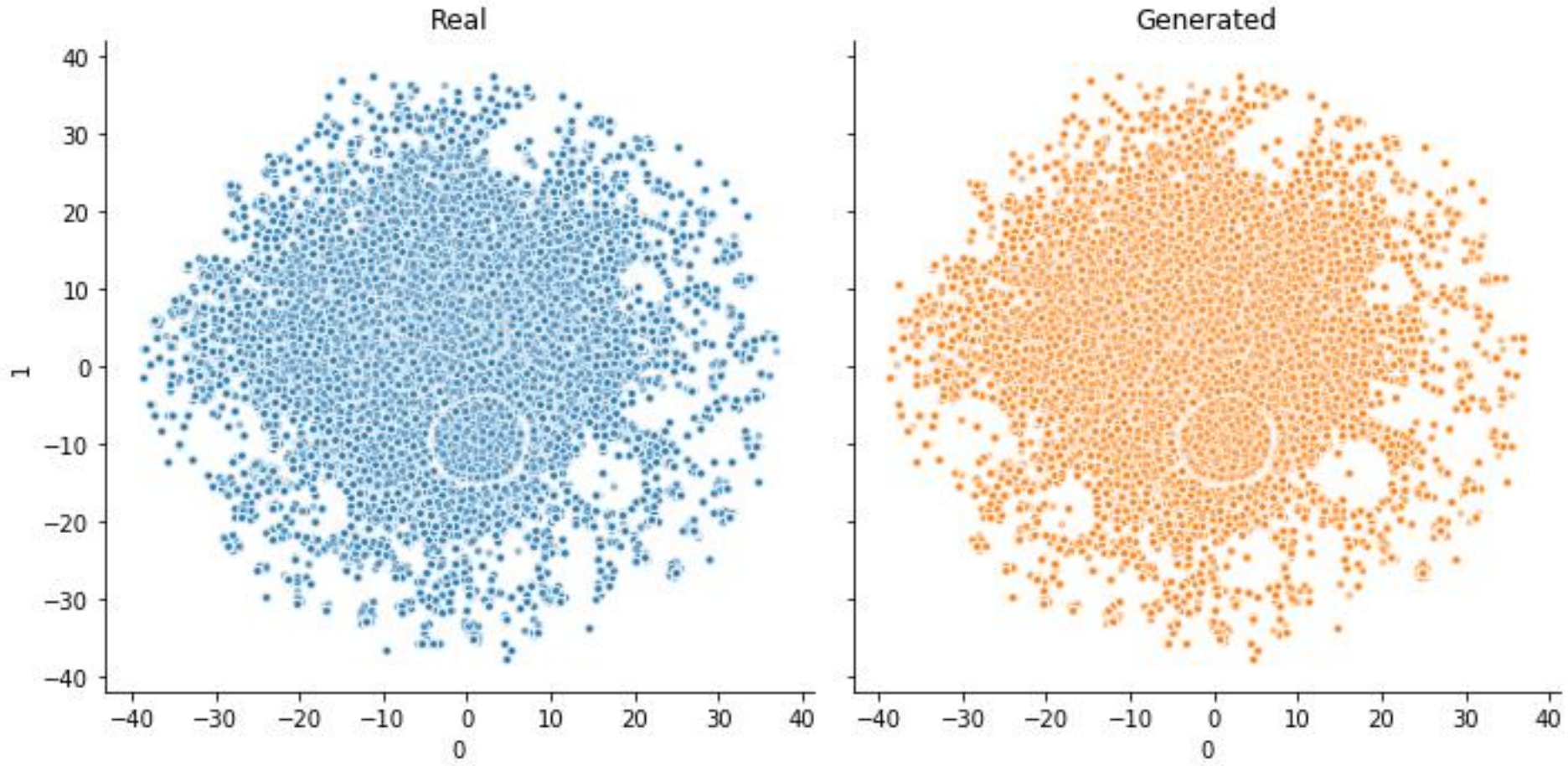
Experimental Results:
Basket Visualization

- Represent baskets as bag-of-product vectors
  - Basket b, products p_i
  - $v_b^{(i)} = 1$ if $p_i \in b$, 0 otherwise

- Plot in 2 dimensions using t-SNE or PCA
- Baskets have similar vectors if the same products tend to occur together
- Similar distributions of generated and real vectors
  - Generator has learned product associations

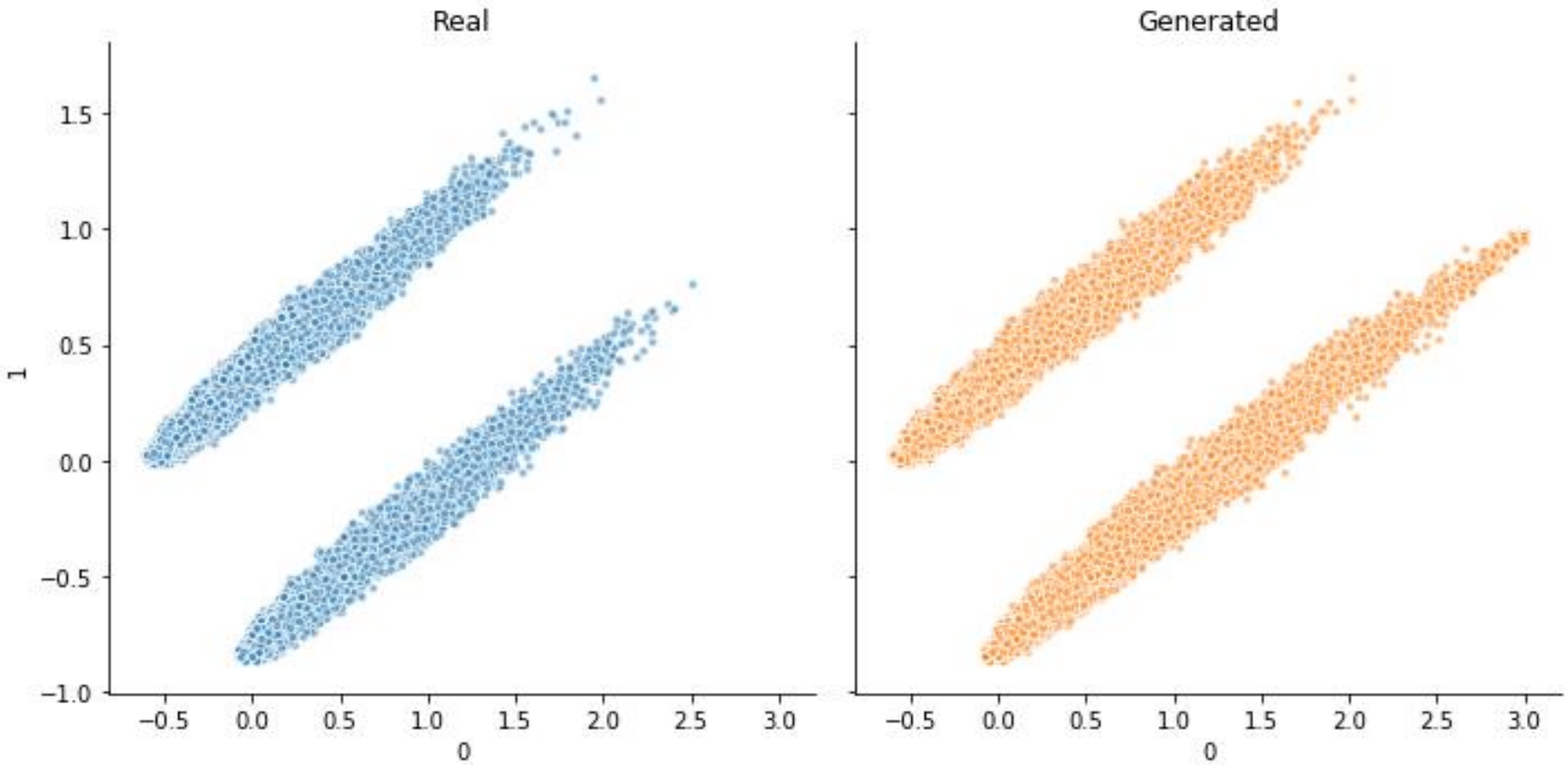
Experimental Results:
Basket Visualization

T-SNE representation



Experimental Results:
Basket Visualization

PCA representation



Experimental Results:
Generated and Real Basket Separability

- Label real baskets with 0 and generated with 1
- Represent baskets as vectors
  - Bag-of-products category: Products in the same category are considered the same
  - Bag-of-products subcategory: Products in the same subcategory are considered the same
  - Basket embedding (sku-level): Take mean embedding of all products in the basket

Experimental Results:
Generated and Real Basket Separability

- Train a classifier to separate real and generated baskets
- Good generator would lead to a classification accuracy of 50%

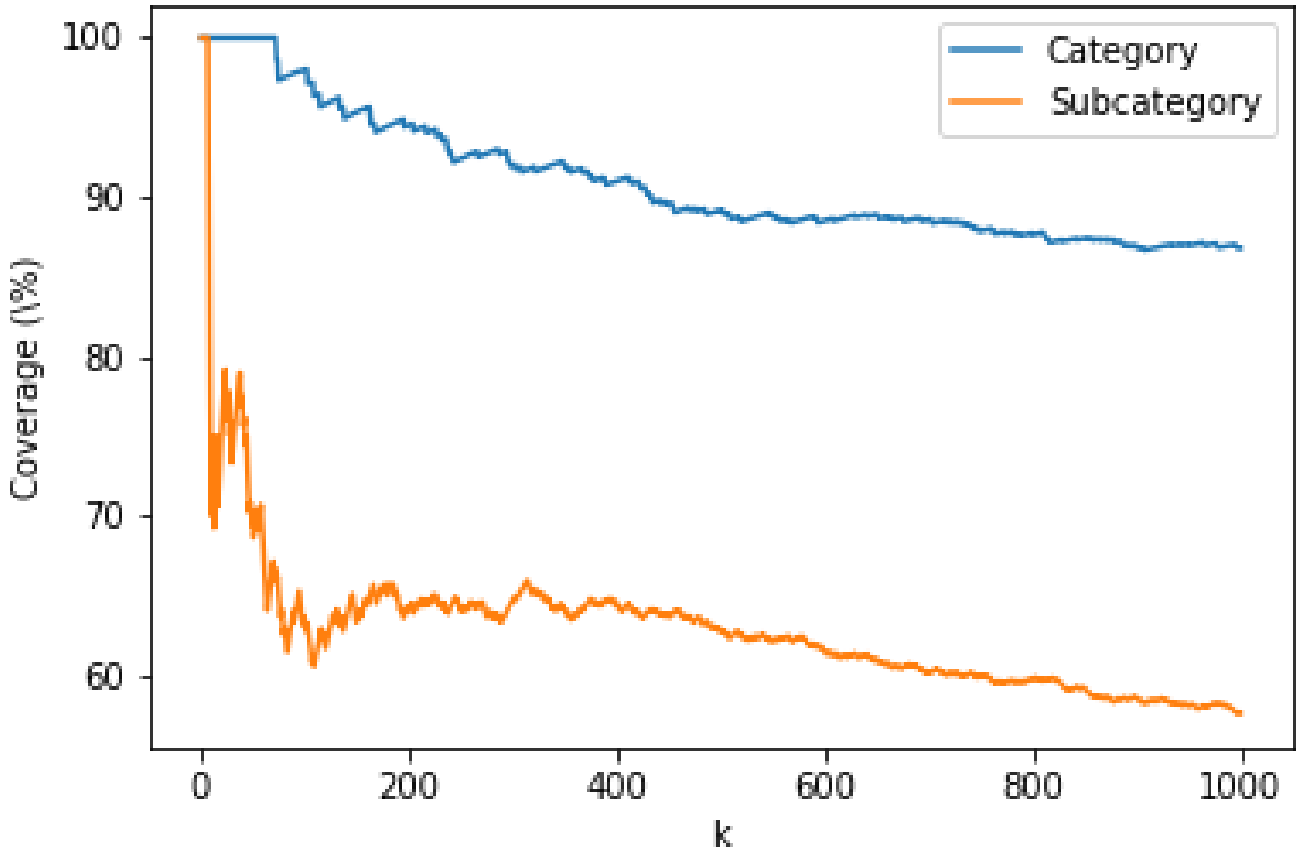
Basket Representation	Classification Accuracy
Bag-of-products category	0.634
Bag-of-products subcategory	0.663
Basket embedding sku-level	0.704

Experimental Results:
Sequential Pattern Mining

- Extract subsequences of items from basket sequences
- Support: fraction of customers containing the subsequence
- Indicates common behavioural trends
- Eg.
- Generated basket sequences should reproduce these trends

Experimental Results:
Sequential Pattern Mining

- Extract top k patterns from the real data
- Compute fraction that are also found in the generated data



Conclusion

- New method of generating sequences of customer baskets
- Conditional GAN to output basket and LSTM to update customer state
- Can be useful for understanding customer behaviour
  - Predict future needs
  - Recommend items

THANK YOU !

QUESTIONS?