Beggars Can't Be Choosers: Augmenting Sparse Data for Embedding-Based Product Recommendations in Retail Stores

Matthias Wölbitsch
Detego GmbH
Graz, Austria
m.woelbitsch@detego.com

Simon Walk
Detego GmbH
Graz, Austria
s.walk@detego.com

Michael Goller
Detego GmbH
Graz, Austria
m.goller@detego.com

Denis Helic
ISDS, Graz University of Technology
Graz, Austria
dhelic@tugraz.at

ABSTRACT

Recommender systems are an essential component in many e-commerce platforms to drive sales and guide customers when exploring new products. With the increasing adoption of RFID technology in traditional brick-and-mortar stores, for example, in the form of smart fitting rooms that allow to display recommendations in the integrated mirror, retailers have only recently started to tap into existing product recommendation algorithms. However, due to limited data availability as well as sparsity, for example due to assortments adapted for different demographics, traditional retailers largely struggle to leverage this technology. In this paper we extend the state-of-the-art embedding-based recommender approach prod2vec by processing information about co-purchased products (i.e., shopping baskets) in retail stores. By adding point-of-sale information to shopping baskets we are able to provide recommendations aimed at individual stores, without having to maintain separate models for each location. Furthermore, we experiment with data augmentation methods to overcome the imposed limitations of the available data, and are able to increase the quality of the computed recommendations by more than 6.9%.

CCS CONCEPTS

• **Information systems → Recommender systems**.

KEYWORDS

recommender; prod2vec; retail industry; shopping baskets

ACM Reference Format:
Matthias Wölbitsch, Simon Walk, Michael Goller, and Denis Helic. 2019. Beggars Can't Be Choosers: Augmenting Sparse Data for Embedding-Based Product Recommendations in Retail Stores. In *27th Conference on User Modeling, Adaptation and Personalization (UMAP '19), June 9–12, 2019, Larnaca, Cyprus.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3320435.3320454

UMAP '19, June 9–12, 2019, Larnaca, Cyprus

ACM ISBN 978-1-4503-6021-0/19/06.
https://doi.org/10.1145/3320435.3320454

1 INTRODUCTION

Recommender systems are omnipresent tools to navigate and explore vast media and product catalogues. Specifically, in the area of e-commerce, recommender systems are often an integral component of the business concepts of successful online retailers, as they present personalized up- and cross-selling options to their customers, increasing revenue and business value.

Problem. However, while online retailers are able to exploit the advantages of recommender systems, traditional retailers (i.e., brick-and-mortar stores) struggle to leverage the potentials of this technology for their business. Although integration of such systems in stores becomes increasingly more feasible due to advances in ubiquitous computing (e.g., smart fitting rooms, which can detect products brought into the changing booth and display recommendations in the mirror) a wide adoption of recommender technology in retail stores is still lacking. The reasons for this are manifold.

First, state-of-the-art recommendation algorithms, such as collaborative filtering, typically require large amounts of customer purchase histories to be able to recommend products. To collect such data, some retailers adopt the concept of loyalty cards, enticing customers with special offers, which allow for customer-purchase association. However, such loyalty programs are often only viable for large retail chains, due to the resulting organizational overhead.

Second, traditional retailers—particularly fashion retailers—often struggle with limited and very sparse data. This is due to small, yet diverse, and fast changing product assortments. For example, different products and inventory sizes are associated with different types of stores (e.g., outlets vs. flagship stores) and with different demographics, for example due to regional differences, leading to limited and sparse data for computing recommendations. Complementary data sources, such as product views, typically used by (small-scale) online retailers to mitigate these problems, are not available for traditional brick-and-mortar retailers either.

Third, alternatives to collaborative filtering, such as content-based approaches, which generate recommendations solely based on the similarity of product properties (e.g., textual descriptions or pictures), require retailers to keep and maintain detailed databases of product information of all products in their inventory.

Approach. An alternative recommender approach is the embedding-based prod2vec [2, 7, 9, 24, 27] algorithm. Similar to content-based approaches, this algorithm generates recommendations based on

product similarities. However, instead of using product properties, it leverages shopping-baskets as the context in which products co-occur to learn low-dimensional vector representations of products (i.e., embeddings). Subsequently, product similarities are computed with standard vector operations, such as cosine similarity.

To tackle the issues related to the application of recommender systems for traditional retailers we extend prod2vec by (i) adding point-of-sale information (i.e., the city in which a purchase was made), which represents an additional location-based context for product embeddings, and by (ii) applying a novel data augmentation approach, which strategically leverages existing shopping baskets to extend product combinations for training our model.

We evaluate our proposed approach on sales data of a total of 20 fashion retail stores owned by a large, international premium clothing manufacturer. The stores are spread across four different cities in a large and diverse country, each with their own characteristics and demographics. With a wide range of experiments on data from these stores, we are able to improve the quality of the generated product recommendations and outperform six different baselines.

Contributions. First, we demonstrate that the inclusion of point-of-sale information in shopping baskets allows us to compile a common model for all four cities, retaining locality effects. Thus, we minimize organizational overhead as only a single model has to be maintained (compared to one model per city). Second, we describe and evaluate our novel data augmentation technique, which further improves recommendations, without the need to change the underlying prod2vec algorithm. Third, we publish our real-world dataset[1] to enable other researchers to re-create and extend our proposed approach and advance research in the context of recommender systems for traditional retail stores.

We strongly believe that our results represent an important step towards the application of recommender systems for traditional retailers, especially in combination with the progressing adoption of ubiquitous computing devices and the resulting possibilities to present recommendations to customers.

2 RELATED WORK

Recommender Systems for Traditional Retailers. While recommender systems are well researched in the domain of e-commerce, the adaption and integration for traditional retailers is still largely unexplored. Walter et al. [26] provide an overview of the problem from a business and technical perspective. Keller and Raffelsieper [13] propose the *Receipt Horizon*, which describes the boundary of what a retailer can learn about their customers. They extend this boundary by introducing a mobile app that uniquely identifies customers and introduces additional features, such as recommendations. Hanke et al. [11] study the adaptability of recommender systems for smart fitting rooms for fashion retailers, and show that the introduction of additional information (e.g., weather conditions) can be beneficial for recommendation performance. Wong et al. [28] provide recommendations with a rule-based expert system in a fashion store using smart fitting rooms, while Buser [4] concentrates on recommendations in grocery stores.

The majority of these studies focus on the integration of recommender systems in retail stores. In this paper, we extend an existing algorithm to take data sparsity and availability (e.g., due to limited and fast changing product assortments) into account.

Embedding-based Recommendations. In the domain of natural language processing (NLP) embedding-based models, which represent words as low-dimensional vectors, are widely used for various tasks, such as word analogies [15, 20]. A popular approach is word2vec [18], which learns word-embeddings by predicting the context of one word (e.g., its surrounding words). It is based on the distributional hypothesis [22], which states that words that appear in the same context are semantically related. In the field of recommender systems, Barkan and Koenigstein [2] apply word2vec on shopping baskets from the Microsoft Store and data from the Microsoft Xbox Music service. Grbovic at al. [9] adapt word2vec to generate product recommendations (i.e., prod2vec) based on e-mail recipes received by Yahoo Mail users. They propose an extension that embeds users as well to provide user-tailored recommendations. Vasile et al. [25] enhance performance, especially in cold-start scenarios, by introducing additional metadata (e.g., artists of songs for music recommendations) to prod2vec. Trofimov [24] uses browsing sessions containing product views as additional source of information. Other applications include the recommendations of home listings to users of the Airbnb platform [7], or matching advertisements with search queries on Yahoo Search [8].

All of these embedding approaches are situated in an online context. The exception to this is the work of Wan et al. [27], who explore embedding-based recommendations for grocery stores. They aim at providing personalized recommendations for regular costumers by capturing brand loyalty effects (i.e., buying the same products over a period of time). As we do not posses customer loyalty information, we base our approach solely on shopping baskets and complement it with location information. Further, data sparsity is not as prominent in grocery store settings, as such stores usually exhibit a higher turnover of larger and less frequently changing product assortments.

Association Rule Mining. Association rule mining algorithms (e.g., Apriori algorithm [1]) analyze shopping baskets and generate a set of rules such as *if you are buying diapers, then you may also like to buy beer*[2]. These algorithms suffer from high computational complexity due to the evaluation of exponentially increasing numbers of product combinations. Although there are several approaches that tackle these issues [10, 16, 23] association rule mining is still based on product counts, neglecting the context of shopping baskets [3, 9] and is not able to capture latent relationships between products.

In contrast, we adapt an embedding-based approach to capture hidden interactions between products. Moreover, we extend the contexts by including additional location information.

Oversampling Strategies. Oversampling strategies are often used to balance classes in datasets. For example, SMOTE [5] (and its variants) compute new stochastic synthetic examples based on underrepresented samples in the dataset. On the other hand, we leverage existing (subsets of) shopping baskets (i.e., discrete items) to extract additional information. Further, we apply our method in the domain of recommender systems rather than classification problems.

[1] https://github.com/detegoDS/shopping_basket_dataset

[2] http://www.dssresources.com/newsletters/66.php

(a) Point-of-Sale (b) Data Augmentation

Figure 1: Extending and Augmenting Shopping Baskets. In Figure (a) we extend a shopping basket with the city in which the sale occurred, represented as location marker. Figure (b) depicts our data augmentation approach. We construct additional shopping baskets consisting of all pairs of products in a given shopping basket. From the given shopping basket composed of a blue jacket, a red T-shirt and green shoes, we construct three new pairs: {blue jacket, red T-shirt}, {blue jacket, green shoes}, {red T-shirt, green shoes} and add them to the training data.

3 APPROACH

We base our approach on prod2vec [9] to compute d-dimensional vector representations of products. To learn product embeddings we leverage co-occurrence information of products in shopping baskets. More formally, given a set of shopping baskets $\mathcal{B}$, where each shopping basket B consists of products from the set of products P, we maximize the following log-likelihood function:

$$\mathcal{L} = \sum_{B \in \mathcal{B}} \sum_{\substack{p_i, p_j \in B \\ p_i \neq p_j}} \log \Pr(p_j \mid p_i). \tag{1}$$

The conditional probability $\Pr(p_j \mid p_i)$ of observing another product p_j from the same shopping basket (i.e., the context) given the current product p_i (i.e., the target) is defined by the softmax function

$$\Pr(p_j \mid p_i) = \frac{\exp(v_{p_i}^\top \cdot v'_{p_j})}{\sum_{p_k \in P} \exp(v_{p_i}^\top \cdot v'_{p_k})}, \tag{2}$$

where v_p and v'_p denote the input and output vector representations for product p. Inferring the remaining products in the shopping basket based on one product corresponds to the skip-gram architecture [18] (i.e., predicting the context, given a target).

Optimizing this log-likelihood function is computationally expensive as we need to compute the normalization term at each step. The corresponding sum iterates over dot products of the embedding of p_i with embedding of every other product in P (i.e., this computation is linear in the number of products). Therefore, we use a hierarchical softmax [19] to approximate the conditional probabilities, which represents the softmax layer as binary tree allowing the computation of conditional probabilities in logarithmic time.

Point-of-Sale. Different environments and demographics affect purchase behavior of customers of traditional retailers. We address this influence by extending each shopping basket with the information of the city in which a sale occurred (see Figure 1a). Therefore, the proposed extension does not require any adaption to the underlying prod2vec algorithm itself, as we only extend the set of products P with a unique identifier for each city and add the corresponding identifier to the shopping baskets. However, we only use this additional information during training and remove the resulting embeddings for the cities from the vector space afterwards, to avoid recommending cities.

In general, our proposed extension is similar to the doc2vec model [14], which in addition to nearby words also adds the paragraph in which a word occurs to the training data. This allows the algorithm to learn vector representations of documents, which can later be used to predict words for a given document.

Data Augmentation. To tackle the problem of limited data in retail stores we strategically augment the available data. The most important piece of information contained in shopping baskets is which products were bought together (i.e., the co-occurrence of products), which is a pairwise relationship. Therefore, we generate all $\binom{m}{2}$ pairs of products for each shopping basket $B \in \mathcal{B}$ with more than two products, where $m = |B|$ denotes the basket size. We add all generated pairs to the set of training examples for the model (see Figure 1b for an example). Next, we construct product triples from shopping baskets to capture and introduce more complex contexts for shopping baskets with more than three products. We limit our construction of additional shopping baskets to pairs and triples due to increasing (exponential) number of larger shopping baskets that could be generated.

In addition to generating new shopping baskets, we also replicate the existing ones in the training set. We apply various replication strategies depending on the size of a given shopping basket (i.e., similar to oversampling). In particular, we copy a shopping basket B of size m so that it occurs m-times in the training set. Further, we investigate different degrees of shopping basket replication, which increases the number of generated replicas even further. Specifically, in addition to replicating a shopping basket of size m exactly m-times, we also duplicate a given shopping basket (i) $2m$-times, and (ii) $\binom{m}{2}$-times.

Computing Recommendations. We compute recommendations by finding the k nearest neighbors of a product in d-dimensional vector space of the product embeddings. The distance (or similarity) between two product vectors is determined by their Euclidean distance. Note that we have also evaluated cosine similarity as distance metric, but achieved better results using Euclidean distance.

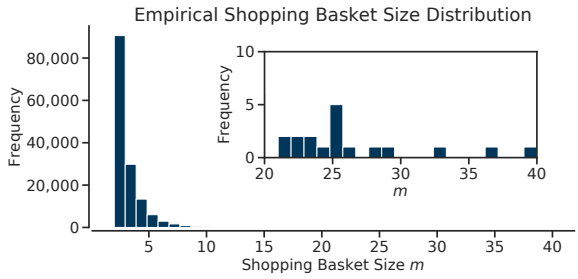

Figure 2: Empirical Shopping Basket Size Distribution. We show sales limited to shopping baskets with two or more products. Most products are sold in pairs or triples. However, we also observe larger shopping baskets (i.e., up to 40 different products; see inset) in our dataset.

4 EXPERIMENTAL SETUP

4.1 Dataset & Preprocessing

Our dataset consists of roughly half a million shopping baskets from 20 stores located in four different cities (Metropolis, Gotham City, Springfield, and Riverdale; anonymized with fictional city names) between November 2016 and December 2018.

Limited Data Availability. We infer product recommendations based on products that were bought together. Therefore, we first discard all shopping baskets consisting only of a single product. This reduces the number of examples available for training to 146,720. Furthermore, we also remove duplicate products from shopping baskets, as we do not gain any additional information for recommendations whenever the products occur multiple times in the same context. This leaves us with approximately 66 purchases per week per store on average, which reflects that goods are typically sold consistently, but in smaller volumes in these stores. In Figure 2 we depict the empirical distribution of shopping basket sizes. Note that these distributions are consistent over all cities.

Data Sparsity. We find more than 17,000 distinct products which are present in at least one shopping basket in our dataset. However, available products differ between cities. The average product assortment similarity, which we calculate using the mean Jaccard similarity coefficient between all city pairs, is 0.51. Hence, we find that the product assortments of stores in the four different cities are adjusted to local conditions and demographics. We list a more detailed breakdown of the key characteristics with respect to the number of shopping baskets and products by city in Table 1.

The number of distinct (unique) shopping baskets is roughly 124,000, which is close to the total number available in our dataset, indicating a heterogeneous buying behavior of customers. The average shopping basket in the dataset contains 2.8 products, which means that most of the time customers only buy few products (cf. Figure 2).

4.2 Experiments

Training Strategies & Baselines. For our experiments we differentiate between two general training strategies. The first is an individual setup, which uses the shopping baskets belonging to one city to fit a model for this specific city. We prefix these models in our work as *individual-cities* approaches. The second is a common setup, which combines the training data from all four cities to fit a single, common model for all of them. We refer to these as *all-cities* models. We evaluate the models of both training strategies for each city separately (i.e., on shopping baskets of the individual cities).

We combine our two training strategies (i.e., *individual-cities* and *all-cities*) with three different baseline approaches to obtain a total of six baselines. In the evaluation step, we compare our approach against these six baselines. In particular, three baseline approaches consist of two simple count-based approaches and a prod2vec model (*prod2vec*) without any extensions. Count-based approaches include (i) a *popularity*, and (ii) a *co-purchase* approach. The *popularity* approach always recommends the k products that appear most often in the training data, whereas the *co-purchase* approach is compiled by counting how often a pair of products co-occurs in shopping baskets. Recommendations in the *co-purchase* approach include k products with the highest co-occurrence for a query product. Combining these three baseline strategies with two training strategies results in the following six baselines: *individual-cities popularity*, *individual-cities co-purchase*, *individual-cities prod2vec*, *all-cities popularity*, *all-cities co-purchase*, and *all-cities prod2vec*.

Point-of-Sale. In the first experiment, we additionally introduce the point-of-sale information (which we refer to as *POS* in our model names) to the *all-cities prod2vec* model. In particular, we explicitly introduce location context to the shopping baskets (see Figure 1a) and compare its performance against our six baselines. We denote this approach *all-cities POS-prod2vec*.

Data Augmentation. In our second experiment we compare several data augmentation approaches. First, we investigate differences in performance for *all-cities prod2vec* models, which we train using our training data as well as generated shopping baskets. To that end, we compile a model which uses the initial shopping baskets and our generated product pairs (i.e., *all-cities pair augmented prod2vec* approach; see Figure 1b). Further, we build an additional model for which we, in addition to product pairs, also add product triples from the training data whenever possible (i.e., for each shopping basket consisting of $m > 3$ products). We denote this approach as *all-cities pair and triple augmented prod2vec*.

Table 1: Dataset Properties. First, we list the number of sales (# shopping baskets) with more than one product for each city (names have been anonymized) and the corresponding percentage, as well as the number of distinct products for each city (# products). Further, we state the mean product overlap for each city compared to all others. The overlap is calculated using Jaccard similarity coefficient (i.e., intersection over union) of the product sets of city pairs.

	# shopping baskets	# products	mean product overlap
Metropolis	31,583 (21.53%)	11,819	0.47
Gotham City	30,269 (20.63%)	9,318	0.51
Springfield	53,108 (36.20%)	11,313	0.54
Riverdale	31,760 (21.65%)	9,597	0.53
Total	146,720 (100.0%)	17,392	

For our shopping basket repetition experiments we duplicate baskets (i) m, (ii) $2m$-times, and (iii) $\binom{m}{2}$-times. Note that the number of replicated shopping baskets in the last experiment coincides with the number of pairs we generate in our proposed *all-cities pair augmented prod2vec* model. We denote approaches fitted with replication as *all-cities x-replicated baskets prod2vec* models, whereas x is a placeholder for the degree of repetition (i.e., m, $2m$, or $\binom{m}{2}$).

Combined Approaches. In our third experiment we study the effects of combining the point-of-sale information and data augmentation. Thus, we first introduce point-of-sale information and then perform data augmentation on the extended shopping baskets. Hence, we obtain models such as *all-cities pair augmented POS-prod2vec* or *all-cities pair and triples augmented POS-prod2vec*.

Finally, we evaluate the performance of an ensemble approach, which is a combination of the best performing models. Vasile et al. [25] already demonstrated in their work that an ensemble can further improve recommendation quality when using embedding-based recommendation algorithms.

Model Parameters. We perform grid search over prod2vec hyperparameters using a 90/5/5 train, validation and test-set split to identify the model configuration which yields the best results for our experiments. Specifically, we evaluate all combinations for the dimension of the product embeddings d starting from 40 to 140 in increments of 20 and training epochs n for the gradient descent algorithm from 80 to 230 epochs. For the best performing configuration we obtain $d = 60$ trained with stochastic gradient descent over $n = 100$ epochs.

Furthermore, for training we randomly downsample high-frequent products from shopping baskets according to the formula proposed for the word2vec model with a threshold t [18]. This reduces the influence of frequently bought 'everyday products', which are sold often but do not provide valuable information for recommendations (e.g., socks). Additionally, we set a minimal total frequency of q for a product and remove all which do not satisfy this condition. We perform grid search over $t \in \{0.1, 0.01, 0.001\}$ and $q \in \{5, 10, 20\}$ and find the best configuration with $t = 0.001$ and $q = 5$. Note that we do not downsample or remove point-of-sale information from shopping baskets.

4.3 Evaluation

With our evaluation we reflect the skip-gram architecture that we used for training our models. Specifically, we calculate a set of k recommendations for each product in a shopping basket of the validation set. We then compare the result set, which is ordered by relevance, against the remaining products in the shopping basket. We repeat this procedure for every product in the shopping basket, so that every product is used as a query product once. If there are no recommendations for a given product (e.g., due to limited support in the training data) we assign the query a score of 0, regardless of the used evaluation metric.

Evaluation Metrics. We evaluate each result set using *recall at k* ($Recall_k$), which is defined as the ratio between the number of relevant products in the result set of recommended products of size k to the total number of relevant products. Hence, using this metric we evaluate to what extent the result set contains the relevant products for a given query product of a shopping basket.

However, $Recall_k$ is not affected by the order in which recommendation are reported, which is an important requirement in real-world recommendation scenarios, as more relevant products should be reported first [6, 17, 21]. Therefore, we also calculate *normalized discounted cumulative gain at k* ($NDCG_k$) [12], which is a rank-based metric that penalizes relevant products at lower ranks in the result set. We calculate the metric using *discounted cumulative gain at k*, which is defined as

$$DCG_k = rel_1 + \sum_{i=2}^{k} \frac{rel_i}{\log_2(i)}, \tag{3}$$

where rel_i is the graded relevance of the recommendation at position i in the result set, which is in our case $rel_i \in \{0, 1\}$, as all products in a shopping basket have the same importance (i.e., $rel_i = 1$) and all others are not relevant at all (i.e., $rel_i = 0$). We then calculate the *normalized discounted cumulative gain* by dividing the DCG_k of the result set by the ideal DCG_k for the query (i.e., the result set which contains all relevant products first).

Evaluation Protocol. We evaluate our approach for each experiment by conducting a 10-fold cross-validation on the whole dataset, using the best model configuration determined via hyperparameter search (see Model Parameters in Section 4). Further, we report average values for both metrics over our 10-fold cross-validation iterations.

We calculate a set of $k = 20$ recommendations for each product in a shopping basket. We have selected k so that the number of resulting recommendations are small enough to be displayed, for example, in the mirror of a smart fitting room. Note that scores for $NDCG_{k=20}$ as well as $Recall_{k=20}$ for shopping baskets larger than 20 products are penalized, as not all potentially relevant products can be obtained. However, as there is only a very limited number of such cases (i.e., 18 of 146,720 in the entire dataset) the impact on the overall results is negligible.

5 RESULTS & DISCUSSION

5.1 Comparison of Baselines

We start by investigating how recommendation performance varies across cities on the basis of our *all-cities prod2vec* baseline. We find that Riverdale and Springfield yield a higher $NDCG_{k=20}$ of 0.1454 and 0.1387 than Gotham City and Metropolis with scores of 0.1144 and 0.1184. We can also find the same pattern for $Recall_{k=20}$ with values of 0.2117 and 0.2032, compared to 0.1781 and 0.1820. In all other experiments and across all models we observe similar and consistent differences in performance for individual cities. Thus, we henceforth only report mean $NDCG_{k=20}$ and $Recall_{k=20}$ over all four cities.

Comparing the performance in $NDCG_{k=20}$ of the count-based *individual-cities popularity* and *co-purchase* baselines, with the *individual-cities prod2vec* models, we see that the embedding-based approach yields better results. The mean $NDCG_{k=20}$ of the *individual-cities prod2vec* models over all four cities is 0.1022, compared to an $NDCG_{k=20}$ of 0.0467 and 0.0637 for the *popularity* and *co-purchase* baselines (see Table 2 rows (c), (a) and (b)).

Further, while the *all-cities co-purchase* and *prod2vec* models (i.e., one common model for all cities) are able to outperform their

Table 2: Experimental Results. This table depicts $NDCG_{k=20}$ and $Recall_{k=20}$ for all of our performed experiments. The presented figures are average values over a 10-fold cross-validation, including the standard deviation in brackets. The experiments are divided into four groups: baselines (a to f), models that leverage point-of-sale information (g), experiments that use the presented data augmentation strategies (h to k), and experiments that leverage both (l to o). The results indicate that models which leverage both proposed methods outperform all baselines in terms of $NDCG_{k=20}$. Nevertheless, the ensemble model of our approach, combined with the *all-cities co-purchase* baseline, yields the highest $NDCG_{k=20}$.

		mean $NDCG_{k=20}$	mean $Recall_{k=20}$
	Baselines		
(a)	*individual-cities popularity* baselines	0.046733 (0.001815)	0.089706 (0.002766)
(b)	*individual-cities co-purchase* baselines	0.063677 (0.001351)	0.193848 (0.004514)
(c)	*individual-cities prod2vec* baselines	0.102156 (0.002850)	0.154447 (0.003785)
(d)	*all-cities popularity* baseline	0.043804 (0.001629)	0.085705 (0.002791)
(e)	*all-cities co-purchase* baseline	0.076699 (0.001568)	0.230796 (0.004683)
(f)	*all-cities prod2vec* baseline	0.129225 (0.003455)	0.193732 (0.004691)
	Point-of-Sale Extension		
(g)	*all-cities POS-prod2vec* model	0.132292 (0.003918)	0.196818 (0.004691)
	Data Augmentation		
(h)	*individual-cities pair augmented prod2vec* models	0.105980 (0.003265)	0.158479 (0.004311)
(i)	*all-cities pair and triple augmented prod2vec* model	0.128519 (0.003585)	0.186634 (0.004419)
(j)	*all-cities m-replicated baskets prod2vec* model	0.127961 (0.004206)	0.186863 (0.004872)
(k)	*all-cities pair augmented prod2vec* model	0.133380 (0.003631)	0.197160 (0.004449)
	Combined Approaches		
(l)	*all-cities pair and triples augmented POS-prod2vec* model	0.133274 (0.003477)	0.194069 (0.004403)
(m)	*all-cities m-replicated POS-prod2vec* model	0.133193 (0.003904)	0.194414 (0.004680)
(n)	*all-cities pair augmented POS-prod2vec* model	0.135105 (0.003777)	0.199929 (0.004836)
(o)	*all-cities pair augmented POS-prod2vec & co-purchase* ensemble	0.138177 (0.003603)	0.215207 (0.004947)

individual-cities counterparts, the *popularity* baseline performs better when fitted for each city separately. We can also observe that the *individual-cities prod2vec* baseline is able to beat the *all-cities popularity* and *co-purchase* baselines with an $NDCG_{k=20}$ of 0.0438 and 0.0767. At the same time, we find that the *individual-cities prod2vec* baselines for each city are on average outperformed by the *all-cities prod2vec* baseline, which achieved a mean $NDCG_{k=20}$ of 0.1292 (cf. rows (d), (e), and (f) in Table 2).

Findings. We find small, yet consistent, differences in performance for each city individually. Further, the *all-cities prod2vec* model is the overall best performing baseline in terms of $NDCG_{k=20}$.

Discussion. In our comparison of recommendation performance across cities we observe small differences. One possible explanation for this effect could be imbalanced training samples from the different cities. However, the best performing city in this experiment is Riverdale, which has a similar amount of training data available as Metropolis and Gotham City (see Table 1). Springfield, for which we collected the most training samples, only achieves the second best performance in terms of $NDCG_{k=20}$ and $Recall_{k=20}$. We hypothesize that the actual differences emerge due to local differences in the shopping behavior of customers, which cannot be captured equally well. This assumption is supported by similar $NDCG_{k=20}$ and $Recall_{k=20}$ of Metropolis and Gotham City, two cities located in closer proximity[3] that are therefore also culturally closer than the other ones.

We also see that while *all-cities* models usually perform better, the recommendation of popular products can be improved by using an *individual-cities* approach. This indicates that the differences in top-selling products are more prominent across individual cities.

5.2 Point-of-Sale

By leveraging the proposed point-of-sale information for the *all-cities prod2vec* model, we can further improve the recommendation quality by 2.86% compared to best performing baseline (cf. row (f) and (g) in Table 2). This *all-cities POS-prod2vec* model achieves a mean $NDCG_{k=20}$ of 0.1323 on average over all four cities.

While the point-of-sale information clearly enhances recommendation performance in terms of $NDCG_{k=20}$ (i.e., more relevant products are ranked first) we can see that the *co-purchase* baselines yield higher $Recall_{k=20}$. In particular the *all-cities co-purchase* baseline achieves the best $Recall_{k=20}$ with 0.2308 compared to, for example, the *all-cities POS-prod2vec* model with 0.1968.

Findings. By introducing point-of-sale information into shopping baskets we are able to improve recommendation quality as measured by $NDCG_{k=20}$.

[3]Note that this is not only true in the DC universe, but also in real life.

Discussion. For this experiment we leverage point-of-sale information for calculating recommendations. Intuitively, models trained for each city individually should already capture this information implicitly. However, *all-cities* approaches outperform *individual-cities* approaches, as we already have shown in the previous Section 5.1. We hypothesize that the sparsity of the available training data, which is a substantial problem for traditional brick-and-mortar retailers in general, is favoring the *all-cities* model simply due to more training data being available.

Moreover, by using an *all-cities* model, we can potentially compute recommendations for newly introduced products in a city, if the model has already encountered these products at different locations (see mean product overlap in Table 1). Hence, we can improve the quality of recommendations in these location-based cold start scenarios. Our results, combined with the additional organizational overhead of maintaining a separate model for each city, suggest the usage of *all-cities* models, even for stores located in different geographic locations. Furthermore, we have shown that the point-of-sale extension further improves recommendation performance of *all-cities* models. Hence, our *all-cities POS-prod2vec* approach is able to overcome data sparsity issues to some extent, while providing a more resilient and location-aware model at the same time.

Nevertheless, we can also observe the strengths of the count-based *co-purchase* models. While these approaches are not able to rank the results well, they appear to be generally better suited to find more relevant products. This is clearly highlighted in the differences in their performance as measured by $NDCG_{k=20}$ and $Recall_{k=20}$ compared to the other models.

5.3 Data Augmentation

First, we report the performance of our data augmentation strategies that generate new training data by leveraging existing shopping baskets. Using the *all-cities pair augmented prod2vec* model we can improve recommendation performance from $NDCG_{k=20} = 0.1292$ to $NDCG_{k=20} = 0.1334$ as compared to the *all-cities prod2vec* baseline (cf. rows (f) and (k) in Table 2). However, we observe a decrease in $NDCG_{k=20}$ of -0.43% and an even higher decrease of -3.66% in $Recall_{k=20}$ with the *all-cities pair and triple augmented prod2vec* as compared to the same baseline (cf. rows (f) and (i) in Table 2).

Second, we compare the best approach so far (i.e., *all-cities pair augmented prod2vec*) against the shopping basket replication strategy, which reintroduces already existing training examples. We observe a decrease in $NDCG_{k=20}$ from 0.1334 to 0.1280 for the *all-cities m-replicated baskets prod2vec* model (cf. rows (j) and (k) in Table 2). Using the *all-cities* 2*m-replicated baskets prod2vec* model we obtain an even lower $NDCG_{k=20}$ of 0.1263, while the *all-cities* $\binom{m}{2}$*-replicated baskets prod2vec* approach performs worst with an $NDCG_{k=20}$ 0.1078. We also observe a similar decrease in $Recall_{k=20}$ for all three replication strategies.

Findings. The data augmentation approach, which generates additional training examples by extracting pairs from the initial shopping baskets (i.e., *all-cities pair augmented prod2vec*), performs better than the other augmentation strategies. This approach also shows a consistent improvement of recommendation performance as measured with $NDCG_{k=20}$ compared to our baselines.

Discussion. With the *all-cities pair augmented prod2vec* model, we introduce roughly 420,000 additional training samples. Out of these, around 320,000 pairs are unique, which is a significant increase in the total number when compared to approximately 70,000 unique pairs in our original dataset. The overlap between the initial and newly generated pairs is approximately 20,000 pairs. Hence, with our pair augmentation approach we introduce numerous new training examples, which are originally not present in our dataset.

In contrast to the addition of new training examples in the form of product pairs, our shopping basket replication strategy reintroduces already existing training examples by putting higher weights on the larger baskets. This leads to a more uniform distribution of the basket sizes. However, this approach fails to improve recommendations, as $NDCG_{k=20}$ and $Recall_{k=20}$ both decrease with higher degrees of replication. Thus, simply repeating training samples does not increase the model performance. We hypothesize that the repetition of less-frequent large shopping baskets does not reflect real-word characteristics very well and introduces noise during the training of our model.

On the other hand, our best performing pair-based strategy is not simply introducing existing shopping baskets multiple times but generates new contexts based on existing ones. Hence, new contexts seem to be more important than replication of the existing ones. Nevertheless, adding product pairs in combination with triples from shopping baskets leads to a decrease in $NDCG_{k=20}$, despite the introduction of additional and more extensive contexts. A potential explanation for this complex non-linear phenomenon is that by introducing product triples we potentially introduce noise as well or skew the actual relations between products.

We confirm this hypothesis by another experiment where we exclusively add product triples whenever possible. In this case we achieve an $NDCG_{k=20}$ of 0.1256, which constitutes a performance decline of -5.84% compared to the *all-cities pair augmented prod2vec* model. Note that when adding only triples, we generate 600,000 new shopping baskets (590,000 unique ones). Compared to the initial 30,000 shopping baskets of size three in our dataset (of which the majority is unique), we observe a small overlap of about 2,000 baskets. Hence, we hypothesize that the introduction of such large amounts of triples does not reflect the characteristics of the original dataset, resulting in empirical shopping basket training samples (opposed to newly generated ones) losing their discriminative and predictive power.

Next, we verify the performance improvement of the pair data augmentation strategy in *individual-cities* models. To that end, we train *individual-cities pair augmented prod2vec* models for each city individually. We observe an improvement in recommendation performance in terms of average $NDCG_{k=20}$ over all cities by 3.74% for the *individual-cities pair augmented prod2vec* model compared to the *individual-cities prod2vec* baseline. We observe a similar performance gain in $Recall_{k=20}$ as well (see Table 2 row (h) and (c)).

Finally, we analyze if we can achieve a similar performance improvement by adding product pairs to our count-based approaches. In our additional experiments we observe that adding new pairs to the count-based baselines does not have a positive effect on their performance. For example, with pair-based data augmentation for the *all-cities co-purchase* baseline we observe a slight decrease in performance in terms of $NDCG_{k=20}$ by -1.9%. Following this result,

we conclude that the augmentation of the dataset using product pairs for training enables the prod2vec algorithm to capture particular latent pairwise relationships between products, which in turn seem to play an important role in the calculation of accurate product recommendations. Note that this is also reflected in the fact that the *all-cities prod2vec* model with no augmentation performs better than all discussed alternative augmentation strategies except pair-wise augmentation.

5.4 Combined Approach

Now we report the results of our combined approaches in which we join our point-of-sale and data augmentation strategies. In particular, compared to the *all-cities prod2vec* model we can see an improvement of a *all-cities pair augmented POS-prod2vec* model in terms of $NDCG_{k=20}$ from 0.1292 to 0.1351. We can also observe a similar improvement in $Recall_{k=20}$ (see Table 2 row (f) and (n)). By using additionally generated pairs as well as triples we achieve an $NDCG_{k=20}$ of 0.1333. If we use instances of the extended shopping baskets multiple times for training (i.e,. m-times, $2m$-times and $\binom{m}{2}$-times), performance does not improve, which is evident in decreasing $NDCG_{k=20}$ (0.1332, 0.1324, and 0.1148).

Finally, we report the performance of a *all-cities* ensemble model, which consists of the best performing models in terms of $NDCG_{k=20}$ (i.e., *all-cities pair augmented POS-prod2vec*) and $Recall_{k=20}$ (i.e., *all-cities co-purchase*). We calculate recommendations for the ensemble by appending the fourteen highest co-purchases of a product to the first six recommendations of our *all-cities pair augmented POS-prod2vec* model. We determined the 6/14 split by performing a grid search over all possible value pairs that sum up to $k = 20$. Using this configuration we can see a performance improvement in $NDCG_{k=20}$ of 6.9% compared to the *all-cities prod2vec* model (cf. Table 2 row (n) and (o)). This constitutes the overall best performing approach in our work.

Findings. We show that the proposed data augmentation in combination with point-of-sale information can improve recommendation quality, especially by leveraging co-purchase information as well.

Discussion. Similar to the data augmentation for initial shopping baskets, the pair-based approach works best for location-extended shopping baskets as well. Hence, by introducing explicit city-product pairs to the training data, we are able to improve the quality of the recommendations even further by making use of the latent interactions between a given city and the products that were sold in that particular city. In this way, we obtain recommendations that are better tailored to stores in individual cities.

While the *co-purchase* baseline ranks the recommendations worse than embedding-based models as evident in lower $NDCG_{k=20}$, it achieves a higher $Recall_{k=20}$. We exploit this fact by constructing an ensemble configuration of our best performing *all-cities pair augmented POS-prod2vec* model and the *all-cities co-purchase* baseline. Here, our intuition is to use the well-ranked results from the embedding-based approach and combine them with relevant products supplied by the co-purchase baseline. Thereby, we effectively eliminate irrelevant products of the *all-cities pair augmented POS-prod2vec* model, and further improve recommendation results in terms of $NDCG_{k=20}$. While we obtain better results also in terms of $Recall_{k=20}$, we are still not able to outperform the *all-cities co-purchase* model.

Nevertheless, we show that point-of-sale information and data augmentation are both (i) suitable to tackle their individually outlined challenges, and are (ii) compatible with each other allowing for a unified recommendation framework, which addresses locality effects as well as limited data availability.

6 CONCLUSION & FUTURE WORK

In this paper we presented a recommendation framework which can be applied in the stores of traditional brick-and-mortar retailers. This setting is characterized by limited available and sparse data, for example due to small and fast changing assortments and varying buying behavior of customers across locations.

We based our work on the already established prod2vec model, which only relies on the co-occurrence of products in shopping baskets and not on additional data, such as customer profiles or product properties (e.g., textual descriptions or pictures). Moreover, we showed that our novel data augmentation approach is able to tackle the problem of data sparsity by generating additional shopping baskets based on already existing empirical data. We also presented a point-of-sale extension, which enhances the shopping baskets with additional location-related information. By combining these methods we provided a resilient and accurate recommendation approach that outperforms individual models for each city. Further, we made another improvement by extending our combined recommender with the benefits of a count-based *co-purchase* approach in the form of an ensemble recommender.

So far we only applied our approach on a single real-world fashion retail dataset. However, we are also interested in experimenting with different datasets, settings, and domains. In particular, we believe that when considering different features of retailers (e.g., department stores vs. specialty retailers) in our training data, we can further improve recommendation performance. Additionally, our presented approaches we were not able to match recall of our co-purchase baseline. To this end, we plan to explore different ensemble strategies to achieve both a high $NDCG_k$ and a high $Recall_k$. Additional metadata information (e.g., size, layout, or turnover of stores) could potentially also enhance recommendations and we plan to extend our approach to include such metadata in the training process for our models.

Other ideas for future work include additional investigation of the importance of product pairs for recommender systems training, the inclusion of methods to reduce cold-start issues for novel products in the discussed retail setting (i.e., fallback strategies for new data) as well as a detailed investigation of the observed performance variations across cities.

We strongly believe that the approach and dataset[4] presented in this paper will improve research in recommender systems suited for traditional retailers, especially due to the progression and adaption of new technologies, such as smart fitting rooms or chat bots. Nevertheless, the presented approach could also be adopted in e-commerce settings, as operators of such websites often serve different markets. Recommendations for visitors of these websites could be tailored towards their distinctive demands and buying behavior as well using our proposed methods.

[4] https://github.com/detegoDS/shopping_basket_dataset

REFERENCES

[1] Rakesh Agrawal and Ramakrishnan Srikant. 1994. Fast Algorithms for Mining Association Rules. In *Proceedings of the 20th International Conference on Very Large Data Bases*, Vol. 1215. 487–499.

[2] Oren Barkan and Noam Koenigstein. 2016. Item2vec: Neural Item Embedding for Collaborative Filtering. In *Proceedings of the 26th IEEE International Workshop on Machine Learning for Signal Processing*. 1–6.

[3] Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don't Count, Predict! A Systematic Comparison of Context-counting vs. Context-predicting Semantic Vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, Vol. 1. 238–247.

[4] Daniel Buser. 2007. Context-Based Recommender Systems in Conventional Grocery—An Economic Analysis. In *Proceedings of the 40th Hawaii International Conference on System Sciences*. 168b.

[5] Nitesh Chawla, Kevin Bowyer, Lawrence Hall, and Philip Kegelmeyer. 2002. SMOTE: Synthetic Minority Over-Sampling Technique. *Journal of Artificial Intelligence Research* 16 (2002), 321–357.

[6] Harr Chen and David R Karger. 2006. Less is More: Probabilistic Models for Retrieving Fewer Relevant Documents. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. 429–436.

[7] Mihajlo Grbovic and Haibin Cheng. 2018. Real-time Personalization using Embeddings for Search Ranking at Airbnb. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 311–320.

[8] Mihajlo Grbovic, Nemanja Djuric, Vladan Radosavljevic, Fabrizio Silvestri, Ricardo Baeza-Yates, Andrew Feng, Erik Ordentlich, Lee Yang, and Gavin Owens. 2016. Scalable Semantic Matching of Queries to Ads in Sponsored Search Advertising. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 375–384.

[9] Mihajlo Grbovic, Vladan Radosavljevic, Nemanja Djuric, Narayan Bhamidipati, Jaikit Savla, Varun Bhagwan, and Doug Sharp. 2015. E-commerce in Your Inbox: Product Recommendations at Scale. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1809–1818.

[10] Jiawei Han, Jian Pei, and Yiwen Yin. 2000. Mining Frequent Patterns Without Candidate Generation. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*. 1–12.

[11] Jannis Hanke, Matthias Hauser, Alexander Dürr, and Frédéric Thiesse. 2018. Redefining the Offline Retail Experience: Designing Product Recommendation Systems for Fashion Stores. In *Proceedings of the 26th European Conference on Information Systems*.

[12] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated Gain-based Evaluation of IR Techniques. *ACM Transactions on Information Systems (TOIS)* 20, 4 (2002), 422–446.

[13] Thorben Keller and Matthias Raffelsieper. 2014. Cosibon: An E-commerce Like Platform Enabling Bricks-and-mortar Stores to Use Sophisticated Product Recommender Systems. In *Proceedings of the 8th ACM Conference on Recommender Systems*. 367–368.

[14] Quoc Le and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents. In *Proceedings of the 31st International Conference on International Conference on Machine Learning*. 1188–1196.

[15] Yitan Li, Linli Xu, Fei Tian, Liang Jiang, Xiaowei Zhong, and Enhong Chen. 2015. Word Embedding Revisited: A New Representation Learning and Explicit Matrix Factorization Perspective. In *Proceedings of the 24th International Conference on Artificial Intelligence*. 3650–3656.

[16] Weiyang Lin, Sergio A Alvarez, and Carolina Ruiz. 2002. Efficient Adaptive-support Association Rule Mining for Recommender Systems. *Data Mining and Knowledge Discovery* 6, 1 (2002), 83–105.

[17] Nathan Liu and Qiang Yang. 2008. Eigenrank: a Ranking-oriented Approach to Collaborative Filtering. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. 83–90.

[18] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems*. 3111–3119.

[19] Frederic Morin and Yoshua Bengio. 2005. Hierarchical Probabilistic Neural Network Language Model. In *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, Vol. 5. 246–252.

[20] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*. 1532–1543.

[21] Stephen Robertson. 1977. The Probability Ranking Principle in IR. *Journal of Documentation* 33, 4 (1977), 294–304.

[22] Magnus Sahlgren. 2008. The Distributional Hypothesis. *Italian Journal of Disability Studies* 20 (2008), 33–53.

[23] Ashoka Savasere, Edward Omiecinski, and Shamkant B Navathe. 1995. An Efficient Algorithm for Mining Association Rules in Large Databases. In *Proceedings of the 21th International Conference on Very Large Data Bases*. 432–444.

[24] Ilya Trofimov. 2018. Inferring Complementary Products from Baskets and Browsing Sessions. In *Proceedings of the 2nd Workshop on Intelligent Recommender Systems by Knowledge Transfer and Learning*.

[25] Flavian Vasile, Elena Smirnova, and Alexis Conneau. 2016. Meta-prod2vec: Product Embeddings using Side-information for Recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems*. 225–232.

[26] Frank E. Walter, Stefano Battiston, Mahir Yildirim, and Frank Schweitzer. 2012. Moving Recommender Systems from On-line Commerce to Retail Stores. *Information Systems and e-Business Management* (2012), 367–393.

[27] Mengting Wan, Di Wang, Jie Liu, Paul Bennett, and Julian McAuley. 2018. Representing and Recommending Shopping Baskets with Complementarity, Compatibility and Loyalty. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 1133–1142.

[28] Wing-Keung Wong, SYS Leung, ZX Guo, XH Zeng, and PY Mok. 2012. Intelligent Product Cross-selling System with Radio Frequency Identification Technology for Retailing. *International Journal of Production Economics* 135, 1 (2012), 308–319.