Learning Embeddings for Product Size Recommendations

Kallirroi Dogani*
ASOS.com
London, UK
kallirroi.dogani@asos.com

Matteo Tomassetti*
ASOS.com
London, UK
matteo.tomassetti@asos.com

Sofie De Cnudde
ASOS.com
London, UK
sofiede.cnudde@asos.com

Saúl Vargas
ASOS.com
London, UK
saul.vargassandoval@asos.com

Ben Chamberlain
ASOS.com
London, UK
ben.chamberlain@asos.com

ABSTRACT

Despite significant recent growth in online fashion retail, choosing product sizes remains a major problem for customers. We tackle the problem of size recommendation in fashion e-commerce with the goal of improving customer experience and reducing financial and environmental costs from returned items. We propose a novel size recommendation system that learns a latent space for product sizes using only past purchases and brand information. Key to the success of our model is the application of transfer learning from a brand to a product level. We develop a neural collaborative filtering model that is applicable to every product, without requiring specific customer or product measurements or explicit customer feedback on the purchased sizes, which are not available for most customers or products. Offline experiments using data from a major retailer show improvements of between 4-40 % over the matrix factorisation baseline.

KEYWORDS

Recommender Systems, Representation Learning, Transfer Learning, E-Commerce

ACM Reference Format:
Kallirroi Dogani, Matteo Tomassetti, Sofie De Cnudde, Saúl Vargas, and Ben Chamberlain. 2019. Learning Embeddings for Product Size Recommendations. In *Proceedings of ACM SIGIR Workshop on eCommerce (SIGIR 2019 eCom).* ACM, New York, NY, USA, 9 pages.

1 INTRODUCTION

Providing customers with accurate size guidance is one of the main challenges in the online fashion industry. Since customers can not try garments before purchasing them, e-commerce platforms often adopt free return policies to motivate customers to purchase items regardless of concerns about size. This effectively turns homes into fitting rooms and encourages customers to order multiple sizes of the same product and return the items that do not fit. According to a recent estimate [2], 15-40 % of online purchases are returned, with an even higher average return rate of 30-40 % for fashion products. It is desirable to minimise returns as the process incurs high operational and environmental costs.

The size problem can not be solved by simply mapping between different sizing schemes such as mapping a EUR shoe size 45 to a UK size 11. There are two reasons for this: (1) inconsistent sizes, for example a men's US size 8 shoe is 10 inches for a Nike trainer [3] while an Adidas trainer measures 10.2 inches [1], (2) simple sizes mask the complexity of the underlying products. For instance, a t-shirt will be sold as small, medium or large, but the size is at least seven dimensional* and there is no standardisation of these dimensions, even for a given brand.

Personalised size recommendations provide a general solution to the size and fit problem. However, the development of a size recommendation system is accompanied by a number of challenges, which we address in our model. Firstly, physical measurements of customers and products are generally not available. Secondly, data indicating that a return was due to incorrect sizing is often missing or unreliable, as it is optionally collected from customers without verification. Thirdly, the presence of an additional size variable makes the data sparser than would be expected in the equivalent product recommendations problem. Finally, the existence of different sizing schemes (e.g. EU, UK, US etc.) introduces heterogeneous data, which must be compared in some way.

We propose the Product Size Embedding (PSE) model, which is a neural collaborative filtering approach that learns a latent representation for all the possible size variations of products and customers' sizing preferences using solely purchase data. By doing so we handle problems with missing physical measurements or returns reasons. We map all sizes into a common continuous latent space, which neatly overcomes heterogeneity in sizing schemes and addresses the inconsistency in sizes that would be hard to address with a discrete combinatorial representation†. To deal with sparsity, we first solve the problem at a brand level by accepting the loose assumption that sizing within the same brand is consistent. Then, we transfer this knowledge onto a product level, where sizes of products within the same brand now have separate representations. Our main contributions are:

- A novel size recommendation system that maps sizes into a single latent space without requiring customer or product

*Both authors contributed equally to this research.

*neck circumference, arm circumference, arm length, height, chest circumference, waist circumference, shoulder width

†such as mapping products to a discrete platonic size scale

SIGIR 2019 eCom, July 2019, Paris, France

physical measurements or explicit customers' feedback on returned items (e.g. too big/small). Our model leads to an improvement of between 4-40 % when compared to the matrix factorisation baseline.
- We show that transferring knowledge learned from a higher level (brands) leads to improved and generalised solutions at a lower level (products).
- We introduce a method to filter out multiple personas from our dataset. Our solution is independent of fixed thresholds or empirically-tuned hyperparameters.

The rest of the paper is structured as follows: Section 2 presents previous related work, Section 3 introduces our proposed model and Section 4 describes how we handle accounts used by multiple personas. Finally, in Section 5 we discuss our experiments and the performance of our model.

2 RELATED WORK

The size recommendation problem has been previously studied in [4, 8, 13, 18–20]. Specifically, [18] models the size prediction task as an ordinal regression problem, where the customer and product true sizes are learned by taking their differences and feeding them into a linear model. [19] extends the work of [18] with a Bayesian logit and probit regression model with ordinal categories. The posterior distribution over customer and product true sizes is based on mean-field variational inference with Polya-Gammma augmentation. The Bayesian approach allows the use of priors for handling data sparsity and the computation of confidence intervals for dealing with noisy data. Both [18] and [19] generate ordinal categorical variables based on explicit customer feedback on returned items (e.g. too small, too big or no return). [8] proposes a Bayesian model that learns the joint probability of a customer purchasing a given product size and the resulting return status being either too small, too big or no return. The probability distribution over sizes is conditioned on the return status and the probability over return statuses is modeled as the empirical distribution over the three possible return events along with a Dirichlet prior based on the counts at the brand and category level. [13] learns a latent space for customers and products by applying ordinal regression. A fitness score is computed for each purchase and size ordering is enforced based on customer's feedback on the purchased size (i.e. too small, too big or a good fit). In order to handle class imbalances, metric learning techniques are applied to transform data into a space where purchases of the same class are closer and purchases of different classes are separated by a margin.

There are two additional studies [4, 20] that tackle the size and fit problem. [4] learns latent product features using Word2Vec [12] and feeds them into a Gradient Boosting classifier along with additional product features (e.g physical measurements, colour, etc.). However, additional product features are often difficult to obtain [6]. Finally, [20] extends [4] to the specific case of footwear size recommendations and also proposes a probabilistic graphical approach that exploits brand similarities.

In literature covering the size recommendation problem, multiple approaches have been employed to reduce noise by identifying multiple personas. The approaches vary from using empirically determined thresholds on the range of purchased sizes to more complex statistical models. [4] filters out users where the mean and standard deviation of the purchased sizes exceeds a category-level threshold. [18] uses a hierarchical clustering method where clusters are iteratively merged as long as the standard deviation of the cluster does not exceed an empirically determined threshold. Each persona is then treated as a separate customer in the subsequent prediction problem. An improvement to the latter work is made in [19], where a persona distribution is drawn from a Dirichlet distribution. Latent variables related to the specific persona are then appended to each purchase transaction. Finally, [8] follows a Gaussian kernel density estimation approach which is further refined to a Gaussian mixture model. Two assumptions are made here: (i) the maximum number of personas is fixed at four, and (ii) the case where only one persona is active is deemed more likely. Each identified persona is subsequently retained in the dataset. A similar problem is tackled in literature focused on identifying active household members in online rental services [5]. Contextual variables such as day of week or time of day are used to identify which member is responsible for which actions and which member is active at a certain point in time.

3 THE PRODUCT SIZE EMBEDDING MODEL

The Product Size Embedding (PSE) model follows a neural collaborative filtering approach to learn embeddings for each product-size combination. The main advantage of the PSE over related latent variable models (e.g. [13]) is that it does not rely on noisy and sparse customer feedback on the returned items (i.e. customers optionally reporting that the item was too big / small). Instead, only implicit signals are used; the products that are purchased and the subset that are returned.

Collaborative filtering [9, 17] uses customer-product interactions and is based on the assumption that customers buying similar products have similar tastes. This principle naturally translates into the size and fit domain as "customers with similar body shapes tend to buy clothes in similar sizes". Matrix factorisation approaches, such as the one proposed by Hu et al. [9], have been proposed to capture the latent taste/preference/style space as reflected by the interactions between customers and products. Matrix factorisation decomposes customer-product interaction matrices into low-rank user and item matrices that represent, respectively, customers and products as vectors in a latent space that captures preferences and styles. Our proposed PSE model similarly represents customers and product sizes in a vector space. However, there are two important differences between our approach and most matrix factorisation approaches. Firstly, we learn a latent space at a product size level instead of at a product level i.e. we have a different vector for every possible size of a product. Secondly, we adopt an asymmetric framework [15] so that users are not represented explicitly, but as the aggregate of the product vectors with which they have interacted. Accordingly, we train different models for each product category (tops, bottoms or shoes), so all trained embeddings belong to the same category and the learned latent space represents the same body part. The asymmetric approach eliminates learning an embedding layer for customers, which greatly reduces the number of parameters. For example, the symmetric approach for menswear

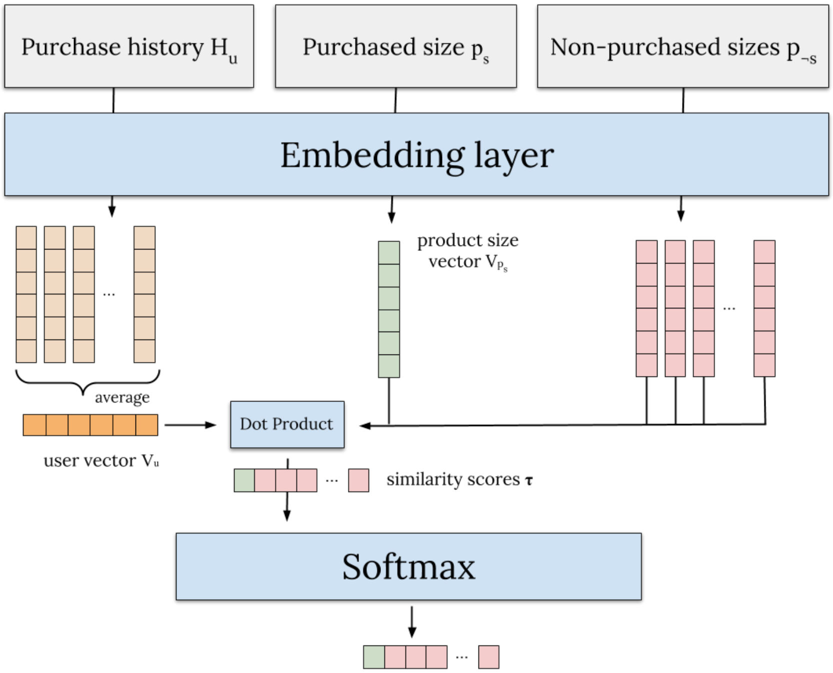


Figure 1: The architecture of the Product Size Embedding model, which is trained independently for each product category (tops, bottoms or shoes) by maximising the dot product between the user vector V_u and the product size vector V_{p_s} of a purchased size p_s. The softmax is computed for each product over all of its possible sizes (i.e. the purchased size p_s and the non-purchased sizes $p_{\neg s}$).

shoes requires ~ 780K product size and ~ 3M customer parameters, therefore the asymmetric model is approximately five times smaller. Another advantage of the asymmetric approach is that the model does not require retraining for new customers since their representations can be inferred from their purchase history. The architecture for the PSE model is shown in Figure 1.

We model size recommendation as a multi-class classification task. Given a user u and a product p, the task is to predict the customers' size in that product, p_s^*. This differs from standard multi-class classification as each product is only available in a small subset of all possible size classes (t-shirts don't come in shoe sizes etc.).

The input to the model is a set of user purchase histories, H_u. For every customer we create a sequence of previously purchased (and not returned) product sizes $\{p_{s1}, p_{s2}, ...p_{sn}\}$. For a sequence on length n, the n^{th} product-size is the target and the previous $n-1$ products are used to construct a customer vector. Each product-size in the history indexes into an embedding matrix using a neural network embedding layer to produce a product-size vector $\mathbf{V}_{p_s} \in \mathbb{R}^k$. User vectors $\mathbf{V}_u \in \mathbb{R}^k$ are constructed by taking the first $n-1$ product-sizes in the H_u, retrieving the associated product-size vectors and taking the mean

$$\mathbf{V}_u = \frac{1}{n-1} \sum_{p_s \in H_{u\setminus n}} \mathbf{V}_{p_s}, \tag{1}$$

where $H_{u\setminus n}$ is the history minus the target size-product. In practice, to increase the amount of training data, for each H_u we will create user and product vectors from all contiguous subsequences of length k where the first (k-1) elements form a customer vector and the k^{th} is the target product-size. The similarity τ between customers and product-sizes is given by the dot product between the user and product vectors

$$\tau_{u,p_s} = \mathbf{V}_u^T \mathbf{V}_{p_s}, \tag{2}$$

and product size probabilities are computed as the softmax of the similarity scores normalised over all sizes of the given product

$$f(\tau)_{u,p_i} = P(s=i|u,p) = \frac{e^{\tau_{u,p_i}}}{\sum_j e^{\tau_{u,p_j}}}, \tag{3}$$

where the index j runs over all possible sizes of product p. To evaluate this softmax we require the product-size vectors for $p_s \forall s$, which are stored in a key-value stored keyed on the product id.

The PSE is trained in Keras using the Adam optimiser [10] with parameters $\alpha = 0.001$, $b_1 = 0.9$, $b_2 = 0.999$ and the categorical cross-entropy loss

$$L = -\sum_{\mathcal{D}} \sum_j t_j \log(f(\tau)_{u,p_j}), \quad t_j = \begin{cases} 1 & \text{if } j = s \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where $\mathcal{D}$ is the extended set of purchase histories and s is the purchased size.

3.1 Transfer from Brands to Products

As we model product-size combinations instead of just products, our product-size interaction matrix is roughly ten times sparser (e.g.

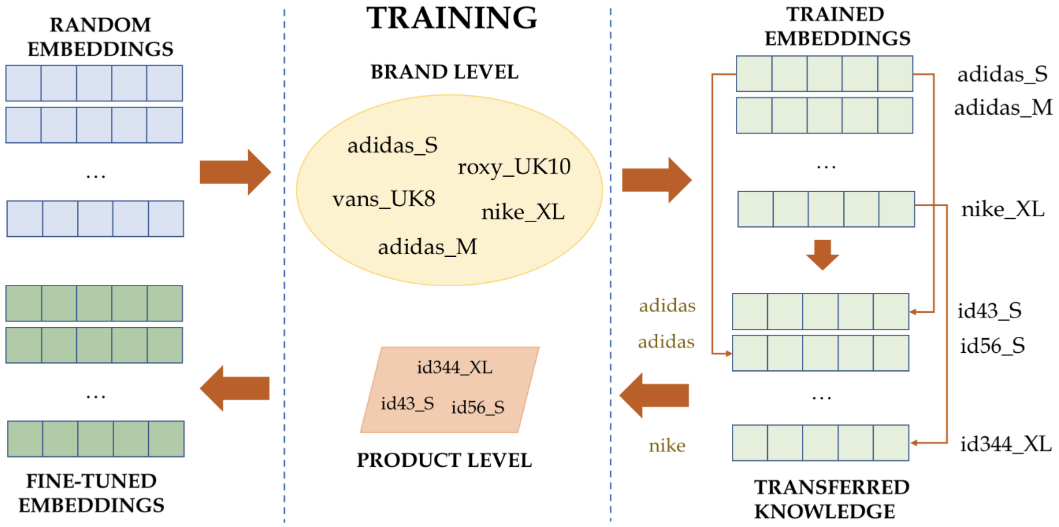


Figure 2: The size embeddings learned at a brand level are used to initialise the size embeddings at a product level.

from ~ 3×10^{-4} to ~ 4×10^{-5} for menswear shoes) than the data used for product recommendations. As a result, learning representations for all possible product-size combinations is challenging. Transfer learning is a popular technique to generalise from small datasets to larger ones [14]. We assume that each brand has consistent sizes and we learn latent representations $\mathbf{V}_{b_s}$ for every combination of brand $b = \{p\}$ and size s. Then, we transfer this knowledge to a product level by initialising

$$\mathbf{V}_{p_s} = \mathbf{V}_{b_s}, \forall p_s \in b_s. \tag{5}$$

As shown in Figure 2, we train the model at a brand size level, then we initialise the product size vectors $\mathbf{V}_{p_s}$ with the trained brand size vectors $\mathbf{V}_{b_s}$ and finally we train the model at a product size level to fine tune the product size vectors. Applying the pre-trained brand size vectors at a product level improves generalisation, boosts performance and leads to faster convergence. In Section 5.3, we demonstrate the improvements transfer learning offers over random initialisation of latent vectors.

4 DETECTING MULTIPLE PERSONAS

A major challenge in the design of recommender systems is identifying accounts that are shared across multiple users. Some services, such as Netflix [7], solve this problem by creating explicit user profiles for each persona. In our work, user profiles are not viable and so we detect multiple personas as a preprocessing step.

To detect multiple personas we employ a Gaussian Mixture Model (GMM) [11] that predicts the number of individuals using an account and identifies each persona's purchases. Our proposed method is independent of assumption-based thresholds or empirically-tuned hyperparameters. When we detect an account with multiple personas, we subsequently remove it from both training and test sets.

Our GMM approach is based on the assumption that the purchases of every persona are centred around a core size. Customers with at least two purchases and with a size difference[‡] larger than one, are potential candidates for the multiple persona detection process. The output of the GMM consists of a mixture of components, each representing a different persona in the purchase history. Each component (or persona) is represented by a Gaussian distribution, whose mean μ corresponds to the persona's core size.

Since the number of personas λ using an account is unknown, we employ the silhouette score s_λ [16] to find the optimal number of mixture components λ_{opt} (see Algorithm 1). The silhouette score is a cluster evaluation metric that measures how well each purchased size is clustered with similar purchased sizes. An $s_\lambda \approx 1$ implies non-overlapping clusters with high density, while $s_\lambda = 0$ points to overlapping clusters.

Algorithm 1 Algorithm for multiple persona detection

```
Input: purchase history $H_u$
Output: $\lambda_{opt}$ persona
  $\lambda \leftarrow 2$
  $s_{\lambda-1} \leftarrow 0$
  $s_\lambda = \mathbf{getSilhouetteScore}(\mathbf{GMM}(H_u, \lambda))$
  while $s_\lambda > s_{\lambda-1}$ and $\min_{i,j=1;i\neq j}^{\lambda} |\mu_i - \mu_j| > 1$ do
    $\lambda = \lambda + 1$
    $s_\lambda \leftarrow \mathbf{getSilhouetteScore}(\mathbf{GMM}(H_u, \lambda))$
  end while
  $\lambda_{opt} = \lambda - 1$
```

The process of identifying multiple personas consists of running the GMM to detect λ personas within H_u and calculating the silhouette score s_λ associated with that mixture. The parameter λ

[‡] We have ordered each sizing scheme from the smallest to the largest size found in our dataset and defined a set of sizing indexes. For examples, the sizing index for the sizing scheme CAT ranges from 0 (3XS) to 25 (8XL). When referring to the size difference between two sizes, we mean their difference when mapped to the sizing index.

Table 1: Example of the output of the multiple persona detection process for womenswear shoes.

Purchase history H_u	Detection
UK3, UK3, UK3.5, UK4, UK4	1 persona
{UK2, UK2}, {UK5, UK5, UK6}	2 personas
{UK2, UK3, UK3, UK3, UK4, UK4}, {UK6, UK6}, {UK9}	3 personas
UK2, UK3, UK4, UK5, UK6, UK6, UK7, UK8, UK9	reseller

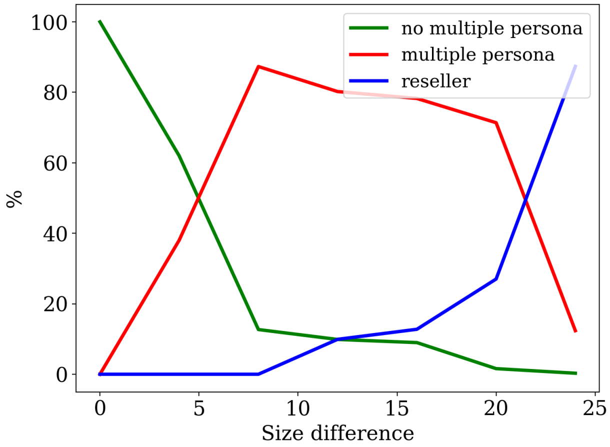


Figure 3: Percentage of multiple persona accounts (red line), reseller accounts (blue line) and no multiple persona accounts (green line) in function of the size difference of the purchases for menswear bottoms.

is iteratively increased as long as (i) s_λ is higher than $s_{\lambda-1}$, and (ii) the core size of each mixture component differs by at least 1 size unit. When the iterative process is finished, λ_{opt} is set to λ and if $\lambda_{opt} > 1$ that customer is identified as buying for multiple personas.

While dealing with the multiple persona problem, two additional issues arise: i) the problem of resellers, and ii) the issue of purchases in multiple sizing schemes. Resellers are customers who purchase products with the intention of reselling them, so it is likely that their purchases cover a wider range of sizes. In that case, a Gaussian mixture model is not suitable for detecting them, as their purchases are not centred around a core size, but instead have a uniform distribution. Therefore, prior to performing multiple persona detection, we eliminate all customers with a uniform purchase history.

To apply the GMM model, we first need to convert all sizes into a single sizing scheme. Since most existing conversion tables are incomplete and inaccurate, we have used the data to approximate size conversions. Specifically, we build a co-purchase matrix per product category between two sizing schemes and we convert sizes according to the highest co-purchase frequency. Note that this conversion is only an approximation for data cleaning purposes and is not used in the final size prediction model.

Table 1 lists examples of purchase histories that are flagged as either multiple personas or resellers.

Table 2: Size range for all sizing schemes.

Sizing Scheme	Size Range
UK	UK2, UK4, ..., UK34
EU	EU30, EU32, ..., EU50
CAT	3XS, ..., 8XL
JNS	W22in L26in, ..., W44in L34in
WST	W22in, ..., W44in
CST	Chest 32in, ..., Chest 56in

The evaluation of the detected multiple personas is similar to evaluating clusters in unsupervised clustering techniques. During the detection process, we calculate the silhouette score, and thus have a built-in evaluation metric that guides the clustering. Figure 3 demonstrates that as the size difference of the purchases increases, the probability of detecting a multiple persona account steadily increases, but it then flattens out and decreases for very large size differences, which indicate a higher probability of detecting a reseller.

5 EXPERIMENTS AND RESULTS

In this section, we first describe the experimental setup, then detail the baselines for comparison and finally present our results. Our experiments are based on data from a major online retailer collected over one year. We have grouped all products into three categories (Tops, Bottoms and Shoes), two genders (menswear (MW) and womenswear (WW)), and six sizing schemes (see Table 2).

The size recommendation problem is solved independently for each product category-gender combination e.g. menswear-tops. Table 3 shows example product types that comprise each product category as well as the supported sizing schemes and high-level statistics. Products originate from a large and diverse network of international suppliers, with thousands of new items added weekly and so in general, physical measurements of products are not available.

5.1 Experimental Setup

Since we solve the size prediction problem separately for each product category, the purchase history H_u has been computed using all previous purchases of customer u from the same product category (i.e. we do not use past purchases of shoes to predict sizes for tops). We exclude any returned products from the purchase history as there is no data specifying whether items are returned due to poor fit or for other reasons.

Table 4 shows examples of the same purchase history computed at different levels. In this case, applying transfer learning from the brand level to the product level means that we initialise the product size vector *id43498_W34inL32in* with the brand size vector *Levis_W34inL32in*.

We divide the dataset for each product category into a training and a test set using an 80:20 split.

5.2 Comparison Methods

We compare the performance of the following personalised methods:

Table 3: Properties and high level statistics of the product categories. WW and MW refer to womenswear and menswear, respectively.

Product Category	Product Types	Sizing Schemes	#users	#products	#brands	% MP	% Resellers
TopsWW	crop tops, hoodies, ...	UK, CAT, EU	3.4M	105.6K	800	9.5%	0.3%
BottomsWW	jeans, leggings, ...	JNS, CAT	1.3M	24.7K	609	4.9%	0.1%
ShoesWW	boots, trainers, ...	UK, EU	1.2M	17.0K	206	3.0%	0.6%
TopsMW	shirts, t-shirts, ...	CAT, CST	1.3M	66.5K	430	5.3%	0.9%
BottomsMW	jeans, chinos, ...	JNS, CAT, WST	840.6K	21.0K	362	3.6%	0.4%
ShoesMW	boots, trainers, ...	UK	391.5K	12.2K	182	2.3%	1.1%

Table 4: The same purchase history generated at different levels.

Level Applied	Purchase History H_u
Brand Level	**Adidas**_L, **Levis**_W34inL32in
Brand & Product Type Level	**Adidas_Shorts**_L, **Levis_Jeans**_W34inL32in
Product Level	**Adidas_Shorts_id3223**_L, **Levis_Jeans_id43498**_W34inL32in

- **MCS-SS**. This method predicts the user's most common size (MCS) given the sizing scheme (SS) of product p. For instance, if $H_u = (id1432\_UK8, id1564\_UK8, id1055\_UK9, id1453\_EU36)$ is the purchase history of user u, this method predicts UK8 for products available in UK sizes and EU36 for products available in EU sizes. If there is a tie, MCS-SS predicts the most recent purchased size.
- **ALS**. This is a symmetric matrix factorisation model optimized through alternating least squares [9].
- **LR**. This is a multi-class Logistic Regression classifier that takes as input the normalised counts of the purchased sizes and one-hot encoded features for the product type, brand and sizing scheme.
- **PSE-B**. Version of the PSE model where the size embeddings are learned at a brand level.
- **PSE-BPT**. Version of the PSE model where the size embeddings are learned at a brand and product type level.
- **PSE**. The size embeddings are learned at a product level.
- **t-PSE-BPT**. The size embeddings are learned at a brand and product type level and the embedding layer is initialised with the latent space learned from PSE-B.
- **t-PSE**. This is our proposed PSE model. The size embeddings are learned at a product level and the embedding layer is initialised with the latent space learned from PSE-B.

We cannot compare our model against other size recommendation algorithms recently published as they require extra data sources that are not always available (i.e. the return reason). Our model is more generic and could be applied to any fashion dataset.

All PSE experiments have been run with a fixed latent space dimension $k = 10$. We have explored the dependency of this parameter on our results and found no statistically significant difference when adopting a higher k (see Fig. 4).

5.3 Results

The results of our experiments are summarised in Table 5. All variations of the PSE model outperform the baselines. We observe that the accuracy increases when the size embeddings are learned at a brand and product type level (PSE-BPT) as opposed to the brand level (PSE-B). However, when latent representations are learned at a product size level (PSE), the accuracy drops for some product categories. If we consider the case of menswear shoes, the number of latent vectors we need to train increases from 1.4K (PSE-B) to 77.9K (PSE), therefore the latent space becomes sparser which makes the model prone to overfitting (Figure 5). To overcome this issue, we have used latent representations learned from PSE-B to initialise the embedding layer in tPSE-BPT and tPSE. The results show that transfer learning improves generalisation and leads to more accurate predictions.

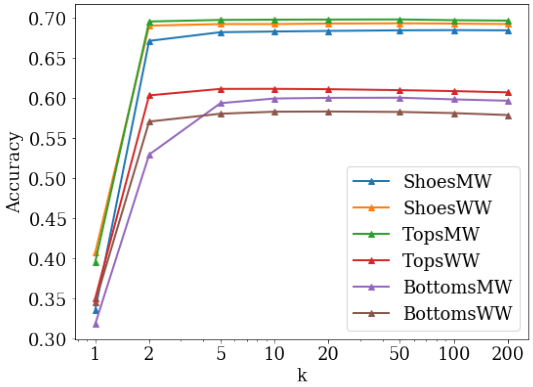


Figure 4: PSE-B accuracy as a function of the latent space dimension, k, for each category. The results are independent of k when $k \geq 10$.

Table 7 shows examples where the tPSE model successfully predicts sizes that are not included in the purchase history, illustrating the benefits of learning latent size representations.

To better understand how tPSE performs in different scenarios, we have evaluated the model on purchase histories of different

Table 5: Accuracy of each tested model for all product categories. The improvement in accuracy for the tPSE model is statistically significant ($^{**}\alpha = 0.01$). WW and MW used in the product categories refer to womenswear and menswear, respectively.

Product Category	MCS-SS	ALS	LR	PSE-B	PSE-BPT	PSE	tPSE-BPT	tPSE
TopsWW	38.917%	60.760%	60.361%	61.175%	61.302%	60.654%	61.294%	**62.286%****
BottomsWW	30.129%	56.440%	57.456%	58.287%	58.446%	58.574%	58.500%	**60.083%****
ShoesWW	63.098%	60.672%	68.354%	69.263%	69.276%	69.518%	69.289%	**70.498%****
TopsMW	64.009%	62.496%	68.689%	69.796%	70.135%	69.542%	70.134%	**70.962%****
BottomsMW	31.893%	52.789%	59.498%	59.964%	60.255%	57.910%	60.290%	**61.992%****
ShoesMW	64.467%	49.160%	68.209%	68.319%	68.612%	65.644%	68.691%	**69.344%****

Table 6: Hitrate@K for tPSE.

Product Category	Hitrate@2	Hitrate@3
TopsWW	88.711%	96.939%
BottomsWW	84.909%	93.835%
ShoesWW	87.529%	94.668%
TopsMW	92.373%	98.315%
BottomsMW	82.485%	90.455%
ShoesMW	86.259%	93.793%

Table 7: Examples of tPSE successfully predicting a size that has not been purchased before.

Purchase History H_u	True Predicted Size
id3455_**UK6.5**, id5637_**UK6**, id4112_**UK6.5**	id9652_**UK7**
id6563_**UK6**, id1463_**UK8**, id3004_**UK6**	id8102_**EU34**

lengths. Figure 6a shows that the accuracy for menswear shoes increases as more items are present in the purchase history. We observe that the accuracy of the model for purchase histories with six or more items is more than 75%. However, this occurs for less than 10% of the data (Figure 6b). The same figure shows that more than 50% of the customers only have one item in their purchase history, which is not sufficient to accurately learn the customer's true size. We observe similar trends for all other product categories.

To confirm that our model does not deviate significantly from the purchased size, we have also evaluated the Hitrate@K, defined as the fraction of times the correct size is within the top K predictions. To retrieve the top K recommended sizes, we rank the predictions based on the similarity scores between the user vector V_u and the product size vectors V_{p_s}. Hitrate@2 ranges between 85-92% for all product categories (Table 6) and can explain cases where customers may be in between two sizes. For instance, both sizes S and M could fit well, but the customer has to pick just one when completing a purchase.

5.4 Analysis on the Latent Space

Figures 7 and 8 show instances of the latent representations mapped onto a 3D space using the t-SNE technique for dimensionality reduction [21]. Specifically, Figure 7 shows the menswear shoes graph constructed by retrieving the closest vectors to *redtape_UK8*. The area around *redtape_UK8* contains brands of size UK8. The neighbourhood in the upper-left corner consists of UK7 sizes, while the area in the bottom-right corner is constructed mainly with UK9 sizes. In the gap between these three big clusters, we observe the half sizes UK7.5 and UK8.5, which show the transitions from the UK8 cluster to the UK7 and UK8 neighbourhood, respectively. In a similar context, Figure 8 shows the latent space of sizes for womenswear tops. The size representations are sorted in ascending order, starting with XS sizes in the upper-right corner and ending with the cluster of XL sizes in the bottom-right corner. Additionally, we observe that same or similar sizes from different sizing schemes (e.g. XS and UK6) are mapped into the same neighbourhoods of the latent space. Both figures confirm the assumption that similar purchased sizes correspond to customers with similar body measurements. Based on this assumption, we can use customer-product interactions to learn a latent space for size representations.

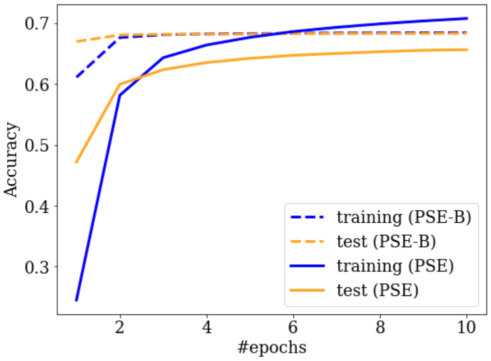
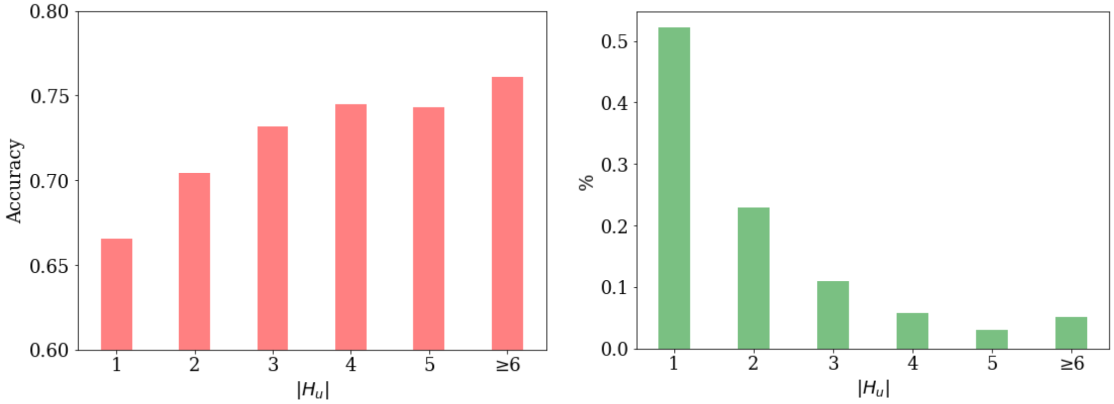


Figure 5: Training (blue lines) and test (orange lines) accuracy as a function of the number of epochs for PSE (solid lines) and PSE-B (dashed lines) in menswear shoes. The model trained at a product level (PSE) starts overfitting after the third epoch, while the model trained at a brand level (PSE-B) is more stable. Similar trends have been observed for the other product categories.

6 CONCLUSION

We introduced the Product Size Embedding (PSE) model, a novel approach to solve the size recommendation problem in fashion e-commerce. The PSE model requires only customer-product interactions and brand information without needing explicit customer



(a) Accuracy of tPSE as a function of the number of items in the purchase history.

(b) Distribution of the length of the purchase history H_u. The dataset is dominated by customers with only one purchased item.

Figure 6: Relation between accuracy and the number of purchases in the purchase history H_u for menswear shoes. Similar trends have been observed for the other product categories.

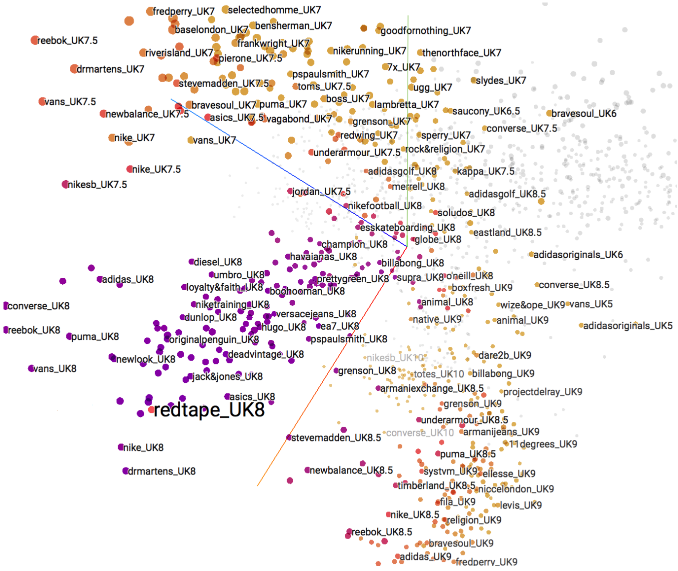


Figure 7: 3D t-SNE projection of the latent space of menswear shoes centred around ***redtape_UK8*****. Purple points are closer to** ***redtape_UK8*** **and represent UK8 or UK8.5 sizes, while orange points are more distant and represent UK7, UK7.5 or UK9 sizes.**

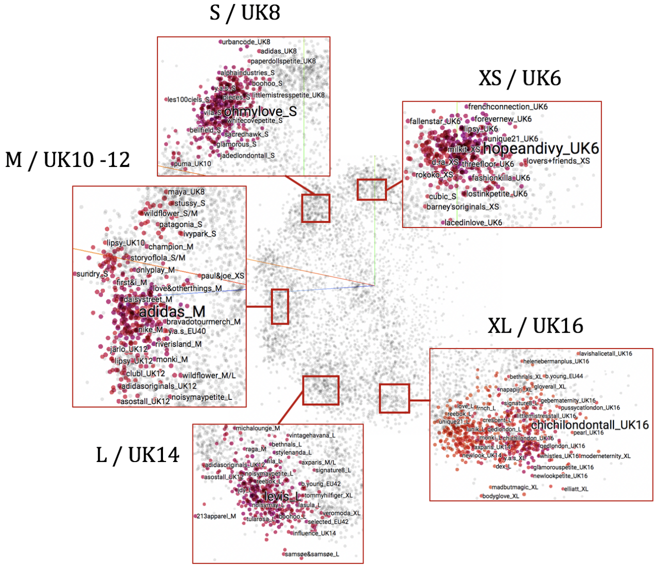


Figure 8: 3D t-SNE projection of the latent space of womenswear tops. The size representations are sorted in ascending order, starting with XS sizes in the upper-right corner and ending in XL sizes in the bottom-right corner. Similar sizes of different sizing schemes are clustered together.

feedback on the returned items (i.e the item was too big or too small). Our offline evaluation on a large-scale e-commerce dataset shows that mapping product sizes into a single latent space leads to more accurate size predictions over a range of different baselines. In addition, we have demonstrated the advantages of transfer learning and how knowledge learned at a brand level boosts the performance of the model at a product level. Finally, we have proposed a technique to identify multiple personas in the purchase history and applied it to reduce the noise in our data.

REFERENCES

[1] 2019. Adidas Size Chart for Men's Shoes | adidas UK. https://www.adidas.co.uk/help/size_charts. Accessed: 2019-01-20.

[2] 2019. Finding a Fix for Retail's Trillion-Dollar Problem: Returns. https://www.cnbc.com/2019/01/10/growing-online-sales-means-more-returns-and-trash-for-landfills.html. Accessed: 2019-01-20.

[3] 2019. Nike.com Size Fit Guide - Men's Shoes. https://www.nike.com/us/en_us/c/size-fit-guide/mens-shoe-sizing-chart. Accessed: 2019-01-20.

[4] G. Mohammed Abdulla and Sumit Borar. 2017. Size Recommendation System for Fashion E-Commerce. In *KDD Workshop on Machine Learning Meets Fashion*.

[5] Pedro G. Campos, Alejandro Bellogin, Fernando Díez, and Iván Cantador. 2012. Time Feature Selection for Identifying Active Household Members. In *Proceedings of the 21st International Conference on Information and Knowledge Management (CIKM '12)*. ACM, pp. 2311–2314.

[6] Ângelo Cardoso, Fabio Daolio, and Saúl Vargas. 2018. Product Characterisation towards Personalisation: Learning Attributes from Unstructured Data to Recommend Fashion Products. In *Proceedings of the 24th International Conference on Knowledge Discovery & Data Mining (KDD '18)*. ACM, pp. 80–89.

[7] Carlos A. Gomez-Uribe and Neil Hunt. 2016. The Netflix Recommender System: Algorithms, Business Value, and Innovation. *ACM Transactions on Management Information Systems (TMIS)* 6, 4 (2016), pp. 13.

[8] Romain Guigourès, Yuen King Ho, Evgenii Koriagin, Abdul-Saboor Sheikh, Urs Bergmann, and Reza Shirvany. 2018. A Hierarchical Bayesian Model for Size Recommendation in Fashion. In *Proceedings of the 12th Conference on Recommender Systems (RecSys '18)*. ACM, pp. 392–396.

[9] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative Filtering for Implicit Feedback Datasets. In *Proceedings of the 8th International Conference on Data Mining (ICDM '08)*. IEEE, pp. 263–272.

[10] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980* (2014).

[11] Bruce G. Lindsay. 1995. *Mixture Models: Theory, Geometry and Applications*. Institute of Mathematical Statistics.

[12] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *arXiv preprint arXiv:1301.3781* (2013).

[13] Rishabh Misra, Mengting Wan, and Julian McAuley. 2018. Decomposing Fit Semantics for Product Size Recommendation in Metric Spaces. In *Proceedings of the 12th Conference on Recommender Systems (RecSys '18)*. ACM, pp. 422–426.

[14] Sinno Jialin Pan and Qiang Yang. 2010. A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering* 22 (2010), 1345–1359.

[15] Arkadiusz Paterek. 2007. Improving Regularised Singular Value Decomposition for Collaborative Filtering. In *Proceedings of KDD Cup and Workshop*. ACM, pp. 5–8.

[16] Peter J. Rousseeuw. 1987. Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis. *Journal of Computational and Applied Mathematics* 20, 1 (1987), pp. 53–65.

[17] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-Based Collaborative Filtering Recommendation Algorithms. In *Proceedings of the 10th International Conference on World Wide Web (WWW '01)*. ACM, pp. 285–295.

[18] Vivek Sembium, Rajeev Rastogi, Atul Saroop, and Srujana Merugu. 2017. Recommending Product Sizes to Customers. In *Proceedings of the 11th Conference on Recommender Systems (RecSys '17)*. ACM, pp. 243–250.

[19] Vivek Sembium, Rajeev Rastogi, Lavanya Tekumalla, and Atul Saroop. 2018. Bayesian Models for Product Size Recommendations. In *Proceedings of the 27th World Wide Web Conference (WWW '18)*. ACM, pp. 679–687.

[20] Shreya Singh, G. Mohammed Abdulla, Sumit Borar, and Sagar Arora. 2018. Footwear Size Recommendation System. *arXiv preprint arXiv:1806.11423* (2018).

[21] L.J.P. van der Maaten and G.E. Hinton. 2008. Visualizing High-Dimensional Data Using t-SNE. (2008).