Leveraging Co-browse Information for Explainable Product Clustering in the Attribute Space

Sneh Gupta
sneh.gupta@flipkart.com
Flipkart Internet PVT LTD
Bangalore, India

Naman Kabra
naman.kabra@flipkart.com
Flipkart Internet PVT LTD
Bangalore, India

Suryanaman Chaube
suryanaman.chaube@flipkart.com
Flipkart Internet PVT LTD
Bangalore, India

Mayank Kant
mayank.kant@flipkart.com
Flipkart Internet PVT LTD
Bangalore, India

ABSTRACT

E-commerce platforms have millions of Stock Keeping Units (SKUs) which makes it extremely difficult to manage the inventory and design appropriate selection/pricing strategies. This necessitates aggregating SKUs into larger units known as Merchant Stock Keeping Units (MSKU). Further, designing critical business strategies on these MSKUs requires them to be robust, interpretable, customer-centric and scalable. Most of the existing algorithms in this domain fail to satisfy one or more than one of the above mentioned criteria. In this work, we propose a novel multimodal tree-based product-clustering approach, wherein, we cluster SKUs on the product attribute space supervised by their co-browse information that is readily available. The proposed methodology splits the products by their attribute values and utilizes their co-browse information for identifying the best split. We recursively split our products until we are left with a very few products or similar products constituting different MSKUs. Leveraging tree-based method enables us to incorporate business guardrails and scale up the approach across different verticals and business units. Experimental analyses on real and synthetic datasets show highly interpretable and robust MSKUs having superior cluster quality with respect to other multimodal clustering techniques on this task. More importantly, the proposed methodology enables us to address the cold start problem in E-commerce by assigning new products to MSKUs that have no browsing information available.

1 INTRODUCTION

A Stock Keeping Unit (SKU) is the most granular unit to track the movement of inventory. In e-commerce, the number SKUs could be very large, which makes it difficult to device efficient inventory management policies. It is thus recommended to segregate large inventory of SKUs for efficient demand planning, selection optimisation, competitive pricing and personalization ([9–11]).

One of the most widely accepted criteria for SKU clustering using ABC methods is the annual dollar usage of SKUs ([4, 9, 14, 15]). However, SKU sales data is sparse as many SKUs do not have any sales data. This makes it difficult to group them with respect to SKU importance using traditional techniques like ABC classification. [2] proposed integrating fuzzy concepts with inventory data to device an analytical hierarchical process for SKU clustering. These strategies, however, are not effective for large verticals of SKUs due to sparsity of sales data. Recently, brand clustering and SKU level clustering conditioned on brands have been investigated by [18] on a latent attribute space. This strategy involves huge number of model parameters which makes it difficult to scale across verticals.

In a fast paced e-commerce environment, new SKUs are frequently added to the inventory to cater to customer needs. These SKUs do not have any browsing or sales information and the only way to assign them to an MSKU is through their attributes. Hence, SKU attributes are critical to creating MSKUs and should generalise well to products that do not have any browsing information. Also, with feature rich web-data (customer shopping) being easily available on E-commerce platforms nowadays, focus has shifted towards using customers' point of view on SKUs to discover intrinsic SKU groupings (see [7]). While SKUs can be grouped based on attribute similarity, they can also be grouped with respect to customers' perceptions on SKU groups: such as SKU group comprising similar brands, similar sizes and shapes, similar colors and patterns etc. Customer perceptions on SKU groups are well reflected on the browsing pattern, eg. similar SKUs being co-browsed by the customers in the same session. This enables scaling up the approach to a large number of verticals as no manually labelled data is required. Additionally, the feature rich information on customers' point of view and SKU attributes can be efficiently utilized to discover SKU grouping for a much larger class of SKUs.

Interpretability, along with adherence to business guardrails, is crucial to designing business-critical strategies and their actionability on the MSKUs. Decision trees were frequently used in previous works where interpretability was prioritised (see [1, 13]). However in these works, only a single modality of information was considered. Combining two modalities of information while clustering is well recognized problem in semi supervised clustering [3]. Recently, semi-supervised clustering on image data has been proposed [12] where users provide pairwise constraints on images as well as the reasoning via image attributes, and both these pieces of information were used for clustering. While this approach helps in incorporating customers' perception, it is a very manual process, making it non-scalable and hence not well suited for SKU clustering. Authors in [6] were able to achieve state of the art performance on a semi-supervised hierarchical clustering task. They proposed

KDD '22, 2022, First Content Understanding and Generation for E-commerce

using cluster-wise tolerance based pair-wise constraints (must-link and cannot-link constraints) to supervise the formed clusters. Our problem could also be framed as an attributed graph clustering problem where the vertices represent SKUs and co-browse score defines the edges. In this domain, I-Louvain [5] acheived state-of-art performance by allowing partition of vertices exploiting both topological structure of graph and vertex attributes. In essence, the proposed strategies in [5, 6] are quite comparable with our MSKU clustering strategy with some straightforward modifications.

We propose, herein, a novel tree-based algorithm denoted as Dual Modality based Decision Tree Clustering(DMDTC) for clustering SKUs/products into MSKUs by bringing together these two information modalities. i.e., SKU similarity with respect to their attributes and customers co-browse data. Node splits are defined on the product attribute space by optimizing for loss with respect to co-browse scores. This helps us in creating MSKUs which are interpretable, customer centric, scalable across verticals and generalisable to new SKUs. More importantly, with the learned attribute space, we are able to assign new products/SKU's to their respective MSKU's, thus addressing a major problem of cold-start assignment in E-commerce. We evaluate our framework on different experimental datasets - two datasets comprising Women Ethnic and Backpack verticals and a synthetic dataset for validation of our methodology. We also compare its performance against algorithms described in [6] and [5] and show that DMDTC is able to achieve better or comparable cluster quality scores on real as well as synthetic datasets.

2 METHODOLOGY

Consider N SKUs within a vertical denoted by $P_1, P_2, \ldots, P_N$. For each of the SKU's, we have information pertaining to a number of categorical and continuous attributes. Apart from the SKU attribute data, we have data of user's browsing history over different sessions. We define a user session as the presence of a user in the platform with a unique IP address who has not visited our platform recently. The proposed clustering strategy starts with considering optimal hierarchical splits of SKU sets into subsets based on the continuous and categorical attribute values. Information on co-browsing pairs of SKUs within same session is leveraged in supervising the splits. Details on preprocessing have been provided in the Supporting Information File (SIF).

2.1 Attribute Information and Cobrowse Score

Let $freq(P_i)$ denote the number of times ith SKU is browsed in the platform within the fixed time interval of consideration. Also, denote $freq(P_i, P_j)$ as the number of times two SKUs P_i and P_j are browsed within a single session. This co-browse information is used to define a score which is the basis of supervising clustering splits. The co-browse score for a pair of SKUs is defined as the likelihood of the pair being viewed in a single user session. Hence in terms of browsing frequency, we define the co-browse score as :

$$cobrowse\_score(P_i, P_j) = \frac{freq(P_i, P_j)^2}{freq(P_i) * freq(P_j)}. \quad (1)$$

Therefore, SKUs which are similar from customer's point of view and hence getting co-browsed, have high co-browse scores. In E-commerce, many SKUs are infrequently browsed and hence most pairs of SKUs do not comprise the co-browse data. Typically, only around 5% of pairs have a co-browse score defined.

2.2 Cluster Similarity

Inter_Cluster_Similarity(ICS) is used to measure similarity between two sets of SKUs. Denoting $|S|$ as the number of distinct elements present in a SKU set, S, for any two sets of SKUs $S1$ and $S2$ the similarity score is defined as:

$$ICS(S1, S2) = \frac{\sum_{P_1 \in S1} \sum_{P_2 \in S2} cobrowse\_Score(P_1, P_2)}{|S1| * |S2|}. \quad (2)$$

Self_Cluster_Similarity(SCS) is used to measure similarity within the same set of SKUs. For given SKUs set S, the score is defined as:

$$SCS(S) = \frac{\sum_{P_1 \in S} \sum_{P_2 \in S - P_1} cobrowse\_Score(P_1, P_2)}{|S| * |S - 1|}. \quad (3)$$

2.3 Business Guardrails

In E-commerce, there are scenarios, wherein, a small number of SKUs are heavily co-browsed with each other but not with any other SKUs. We define these kind of SKUs as Isolated SKUs. Isolated SKUs would generally have a very high SCS score and a very low ICS score (defined in equations 3 and 2). Therefore, implementing clustering in the absence of any business guardrails would lead to each Isolated SKU getting assigned to a separate (MSKU) cluster. This makes MSKUs unusable for business use cases as it is not very feasible to devise business strategies on a multitude of small MSKUs. Our approach enables easy incorporation of business guardrails into the algorithm like minimum number of SKUs within an MSKU, minimum revenue contribution from each MSKU etc.

2.4 Algorithm

We propose a tree-based clustering approach that splits SKUs into two subsets based on their attribute values, supervised by the best split score. If a binary split segregates an SKU set S into two subsets $S1$ and $S2$, the split score of this operation is defined based on the following equation:

$$split\_score(S1, S2) = \frac{ICS(S1, S2)}{SCS(S1) + SCS(S2)}. \quad (4)$$

We recursively split the SKUs as long as we get good enough splits. A split is considered good if both the subsets satisfy business guardrails and the obtained best *split_score* is less than a user defined threshold. Alternatively, users can tune the maximum depth of the tree to limit the number of attributes used for clustering. Following this approach, we obtain a tree structure with leaf nodes constituting the final clusters (MSKUs).

The implementations of the proposed strategy contains three main functions as described in details in **Algorithm 1**:

(1) In **line no. [1, 14]** get_splits(m) returns possible splits for m'th attribute. It should be kept in mind that the possible splits for categorical and continuous attributes are different as mentioned in the **Preprocessing** step. WLOG first M_1 attributes are categorical.
(2) UDC checks whether both the subsets satisfy user defined constraints. This function can be different for different use cases. (Used in **line 24**)

Algorithm 1 Proposed MSKU Clustering

```
1: function GET_SPLITS(m, P)
2:     splits ← []
3:     attr_values ← {}
4:     for i ← 1 to len(P) do
5:         attr_values ←(insert) attr_map(P[i], m)
6:     if m ≤ M_1 then
7:         splits ← all subsets of attr_values
8:         for split s in splits do
9:             if attr_values[0] not in s then
10:                delete s
11:    else
12:        sort attr_values
13:        for i ← 1 to len(attr_values) - 1 do
14:            splits[i] ← attr_values[1 : i]
       return splits
15: procedure DMDTC(P)                                   ▷
16:    attr_splits ← []
17:    for m ← 1 to M do
18:        attr_splits[m] ← get_splits(m, P)
19:    split_scores ← {}
20:    for m ← 1 to M do
21:        for split s in attr_splits[m] do
22:            subtree1 ← SKUs whose m'th attribute value in s
23:            subtree2 ← P - subtree1
24:            if UDC(subtree1, subtree2) = False then
25:                continue
26:            intra_score ← SCS(subtree1) +SCS(subtree2)
27:            split_scores[m, s] ← ICS(subtree1,subtree2) / intra_score
28:    if min(split_scores) ≤ Threshold then
29:        m, s ← argmin(split_scores)
30:        subtree1 ← SKUs whose m'th attribute value in s
31:        DMDTC(subtree1)
32:        DMDTC(P - subtree1)
33:    else
34:        Define P as new cluster
35:
```

(3) In **line no. [15, 35]** DMDTC(P) returns clusters from P SKUs. Here, we first identify all the possible splits and check their corresponding split_score. We look for those splits that satisfy UDC defined in (2) and their split_score is better than the Threshold. If there exists one such split, we split our SKUs into two subsets and recursively run the algorithm, else we assign SKUs to one cluster.

2.5 Cluster Performance Comparison

Since we are combining information from two modalities (attribute and co-browse), it is fair to compare our algorithm with multimodal/semi-supervised clustering techniques. In the current implementation, we have benchmarked our algorithm against [5, 6].

In [5] , vertices are defined by SKUs and edges between them are defined by their co-browse scores. The objects in [6], on the other hand, are defined by the corresponding attributes and we define must link contraints between them if their cobrowse score is higher than a threshold. In our case, for MSKU clustering, the cannot-link constraint is not defined as in case of [6]. Results are published for the threshold that yielded best test results. To make the clusters meaningful and actionable from a business perspective, we impose a threshold of (minimum) 1% of the total SKUs for each cluster (MSKU). Due to the presence of isolated SKUs, a direct implementation of the algorithms in [6] failed to satisfy this criterion on our dataset.

To discourage this from happening in [6] we introduced a size penalty (α) (denoted by SSAHC1) in the formula to calculate the distance between two clusters and re-define Eq. (9) in [6]. We re-define the distance between two clusters as

$$d(C_i, C_j) = \begin{cases} (AG(C_i, C_j))^2 & \text{if } AG(C_i, C_j) \geq 0 \\ 0 & \text{otherwise} \end{cases}, \quad (5)$$

where $Ag(C_i, C_j) = ||M(C_i) - M(C_j)|| - K(C_i; C_j) - K(C_j; C_i) + \alpha * (n_{Ci} + n_{Cj})$. The final L clusters are denoted by $C_1, C_2 ... C_L$ and $|C_i|$ denotes number of points in cluster C_i. We define the proportion of points in a given cluster by $n_{Ci} = \frac{|C_i|}{\sum_{k=1}^{L} |C_k|}$.

Another variation of the clustering strategy (denoted by SSAHC2) was attempted by initially forming $m(> L)$ clusters and subsequently merging clusters which did not follow business guardrails with the nearest clusters (following business guardrails) to obtain L final clusters. Maximum value of m was chosen to be 20. The final result for [6] (denoted by SSAHC) denotes the maximum clustering score of the two variations.

2.6 Experimental Datasets

We consider a couple of experimental datasets on two disparate SKU verticals for our analyses. The first dataset comprises 1055 SKUs in the Women Ethnic vertical and the other one comprises 866 SKUs in the Backpack vertical. Each SKU in the Women Ethnic dataset has 12 attributes (2 continuous and 10 categorical). The continuous attributes are price and length of the blouse piece, while the categorical variables include the fabric, embroidery type etc. of the respective SKU. Each SKU in the Backpack dataset has 12 attributes (6 continuous and 6 categorical). The continuous attributes include price of the SKU, number of compartments present in a backpack etc. and categorical variables include the brand, type of material etc. of the backpack. Dataset was split randomly into train and test with test size 20%.

Since ground truth labels are not present and distance from cluster centroid cannot be defined for co-browse data, we use Silhouette Index (SIL) as defined in [17] to validate our cluster quality where distance between two SKUs P_i and P_j is defined as: $d(P_i, P_j) = -cobrowse\_score(P_i, P_j)$. Clusters are built using train dataset comprising product attributes supervised via co-browse information while test data points are assigned cluster membership basis their attributes alone (ignoring their co-browse information). The rationale behind above definition is that new SKUs (eg. newly onboarded products on the platform) do not have co-browse information and the only way to assign clusters membership to them is via their attributes.

Table 1: Silhouette index for varying number of clusters(L) for Women Ethnic Dataset.

Women Ethnic Dataset												
Algorithm	L	A+C	A	L	A+C	A	L	A+C	A	L	A+C	A
K-Means*	5	0.51	0.52	10	0.35	0.33	15	0.25	0.23	20	0.17	0.15
SSAHC	5	0.75	0.55	10	0.55	0.43	15	**0.56**	0.30	20	**0.36**	0.30
I-Louvain	5	**0.86**	0.26	10	**0.83**	0.16	15	-	-	20	-	-
DMDTC	5	0.67	**0.64**	10	0.47	**0.49**	15	0.41	**0.38**	20	0.34	**0.32**

Table 2: Silhouette index for varying number of clusters(L) for Backpack Dataset.

Backpack Dataset												
Algorithm	L	A+C	A	L	A+C	A	L	A+C	A	L	A+C	A
K-Means*	5	0.46	0.39	10	0.37	0.27	15	0.21	0.19	20	0.15	0.14
SSAHC	5	0.88	0.73	10	0.64	0.67	15	0.58	0.46	20	0.53	0.45
I-Louvain	5	0.73	0.46	10	**0.83**	0.34	15	**0.85**	0.20	20	**0.87**	0.18
DMDTC	5	**0.91**	**0.90**	10	0.76	**0.80**	15	0.69	**0.67**	20	0.67	**0.60**

* For K-Means only A was considered in case of A+C

2.7 Synthetic Dataset

Since Silhouette score quantifies the cluster quality based entirely on the notion of mean intra-cluster and nearest-cluster distance for each sample, it conveys no information regarding the 'correctness' of clustering, which requires knowledge of ground truth. In order to validate our methodology, we run experiments on a synthetic dataset with 10 clusters comprising 200 data points each. Each point could be considered as a product defined by its attributes that are a mix of continuous and categorical attributes (we consider 10 continuous, 5 categorical and 5 noisy attributes). To make the attributes informative and distinguishable, we assigned 0 and 1 labels to categorical attributes with proportion of either label chosen randomly for each of the attributes (for all clusters). For continuous attributes, we randomly sampled mean for each of the clusters from a normal distribution with unit variance. The points within a cluster were then assigned values sampled from normal distribution with respective cluster's mean and 0.5 variance. Noisy attributes had zero means for all the clusters. Further, any two points P_1 and P_2 were defined to have co-browse scores, based on whether they belong to same or different clusters, as follows:

$$\begin{aligned} cobrowse(P_1, P_2) &\sim \mathcal{N}(\mu = 1,\ \sigma^2 = 1) \forall (P_1 \in S_i, P_2 \in S_j), i = j \\ &\sim \mathcal{N}(\mu = 0,\ \sigma^2 = 1) \forall (P_1 \in S_i, P_2 \in S_j), i \neq j \end{aligned} \tag{6}$$

Above normal distributions with unit and zero means and unit variance ensure that we have distributions on cobrowse scores with higher (lower) average scores for points belonging to same (different) clusters. Fundamentally, this score could be thought of as the (scaled) linkage probability between a pair of points with a higher value, i.e., cobrowse score for points within a cluster. Linkage dropouts with an upper threshold of 20 % were randomly applied to each of the clusters to introduce some stochastisity in the dataset. Since the ground truth in this case is known apriori, i.e. cobrowse scores, it becomes possible to compute the Adjusted Rand Index (ARI) metric [16], which captures the similarity between two clusterings by considering all pairs of samples and counting pairs that are assigned to the same or different clusters in the predicted and true clusterings [8]. For a given set S of n elements and two distinct clusterings of these elements, namely $\{X = X_1, X_2, ..., X_r\}$ and $\{Y = Y_1, Y_2, ..., Y_r\}$, the overlapping elements in X and Y could be described through a contingency table $[n_{ij}]$, with each entry in n_{ij} depicting the number of common objects between X_i and Y_j and a_i and b_j representing the summation of $[n_{ij}]$ along the rows and columns of the table, respectively.

$$\frac{\sum_{ij} \binom{n_{ij}}{2} - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right] \Big/ \binom{n}{2}}{\frac{1}{2}\left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right] - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right] \Big/ \binom{n}{2}} \tag{7}$$

The index typically ranges from 0 to 1, with index value equal to 1 when a partition is identical to the intrinsic/true structure.

3 EXPERIMENTAL RESULTS

3.1 Numerical Performance - Experimental Datasets

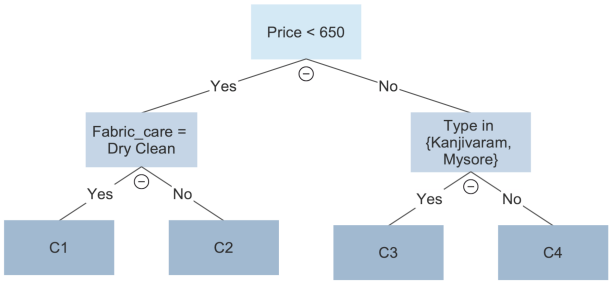

Figure 1: DMDTC Clusters for Women Ethnic vertical (with no. of clusters restricted to 4)

Fig. 1 depicts the MSKUs for all products in the Women Ethnic vertical. For example, the products in C3 MSKU have 'Price >= 650' and 'Type = Kanjivaram or Mysore'. Definitions of other clusters can be derived similarly. The tree-based nature of the algorithm enables easy interpretation of clusters, for every product can be assigned to its respective cluster based entirely on its attribute information (Price, Type, Fabric_care in this case).

Table 3: ARI for number of clusters(L) = 10 on Synthetic Dataset.

		Synthetic Dataset: 5 Noisy			Synthetic Dataset: 15 Noisy		
Algorithm	L	A + C	A	Gen. ARI	A + C	A	Gen. ARI
SSAHC	10	0.58	0.53	0.05	0.1	0.23	0
I-Louvain	10	**0.89**	0.71	0.18	0.73	0.67	0.06
DMDTC	10	0.81	**0.78**	**0.03**	**0.81**	**0.78**	**0.03**

The Silhouette scores for the Women Ethnic and Backpack datasets are presented in Table 1 and Table 2 respectively. We denote the attribute and cobrowse spaces with A and C, respectively. We report the difference in the Silhouette scores in Train (A+C) and Test (A) data as an indicator of how well the clustering strategy generalises to products based only on attributes: $Gen.SIL = Abs(TrainSIL - TestSIL)$. Fig. 2(a-b) show the *Gen.SIL* plotted against the number of clusters. Depending on the business use case, it might be more useful to have fine grained or coarse grained clusters, thus we have compared the algorithms across a diverse range of final number of clusters (L). For I-Louvain the number of clusters cannot be directly varied from the optimal clusters generated by the algorithm. For comparison across a diverse range of L, SKUs from smaller clusters are merged with the remaining clusters on the basis of minimum mean attribute distance from the cluster SKUs.

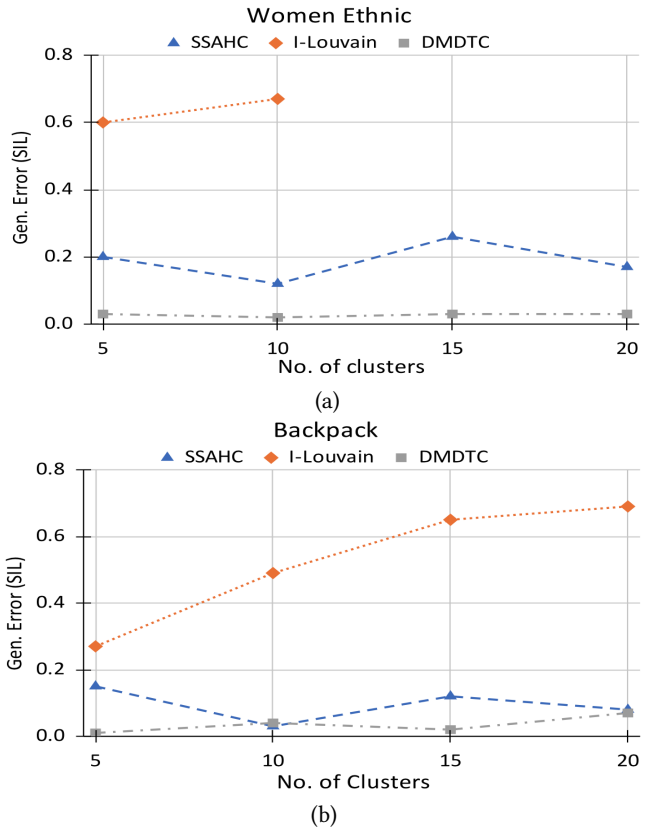

Figure 2: Generalization error across varying number of clusters for (a) Women Ethnic and (b) Backpack verticals

From Tables 1-2, it can be seen that the performance of K-Means is worst for both the verticals, as only a single information modality (attribute) is exploited. In case of I-Louvain, for both the verticals, the generalization error increases with number of clusters while for DMDTC and SSAHC, the *Gen.SIL* is largely constant. DMDTC, however, has the lowest *Gen.SIL* in both the cases. For I-Louvain, the number of clusters cannot be varied directly and hence results are not generated for all values of L. For instance, on the Women Ethnic dataset, the optimal clusters generated with I-Louvain were always less than 15. Thus, we have not published the results for $L = 15$ and $L = 20$.

For both Women Ethnic and Backpack dataset, the DMDTC algorithm performs better with respect to the Test SIL index than both the versions of SSAHC and I-Louvain on all 4 sizes (L) of final clusters as shown in Table 1 and 2. Lower values of *Gen.SIL* indicates robustness of the algorithm as well as simpler splitting criteria compared to other algorithms. A higher test SIL (on A space) across a diverse range of final number of clusters as shown in Fig. 2(a-b) indicates that we are able to create good quality clusters without compromising on interpretability.

In case of SSAHC, the best results were obtained when the cluster tolerance parameter was non-zero. This, along with inferior performance of K-means algorithm re-emphasizes the importance of exploiting multiple information modalities on this clustering task. The DMTDTC algorithm is able to discard less-relevant attributes from customers point of view because splitting through those does not result in a significant gain in split score (unlike SSAHC).

3.2 Numerical Performance - Synthetic Dataset

Table 3 depicts the performance of SSAHC, I-Louvain and DMDTC algorithms on the synthetic dataset for $L = 10$. Clearly, the performance of both DMDTC and I-Louvain is significantly better than SSAHC while the generalization error is lowest in case of DMDTC (Table 3). Fig. 3 shows the performance of I-Louvain as the weightage of the Newman modularity (pairwise linkage) in Eq. (2) [5] is varied on the synthetic dataset. It can be inferred that the performance of I-Louvain is highest when the pairwise linkage weight is close to one, i.e. most of the learning happens in the cobrowse space itself.

Fig. 4 depicts the comparison between DMDTC and I-Louvain as the number of attributes is varied from 5 (20 %) to 20 (100 %). For each of the runs, the proportion of noisy attributes was maintained at 25 %. Clearly, the drop in performance (ARI) from (A + C) to A space is pronounced in case of I-Louvain while for DMDTC, there is hardly any performance drop even for smaller number of attributes. Table 3 also depicts that the generalization error in case of DMDTC is lowest even with more noisy attributes introduced (< 0.05 ARI).

3.3 Discussion

The proposed multimodal framework is a significant advancement over previous works on two counts. One, it creates interpretable,

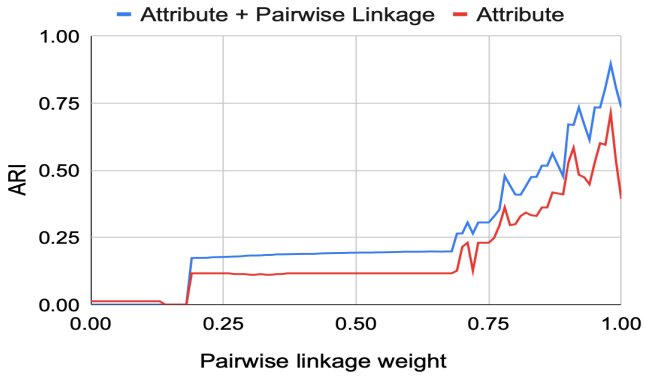

Figure 3: I-Louvain performance against varying weight of the Pairwise Linkage matrix

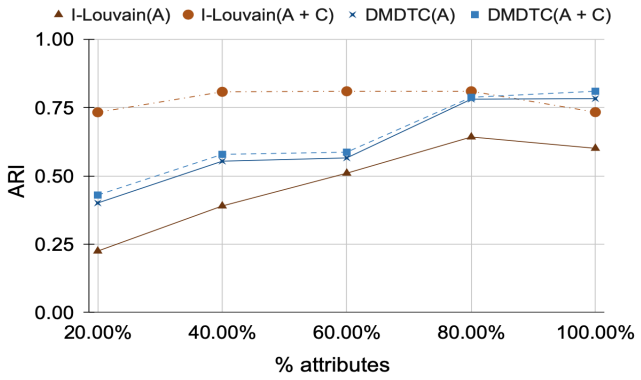

Figure 4: Comparison of DMDTC with I-Louvain as a function of the dimensionality of attribute space

robust, scalable and customizable product clusters while ensuring customer centricity. For instance, the clusters (MSKUs) highlighted in Section 3.1 convey brand as an important attribute and also the similarity perception amongst brands from a user's lens; eg. Safari and American Tourister in Cluster 1 are similar or substitutable brands (based on cobrowse). One can also easily customise the number of MSKUs, vis à vis unit threshold, tree splitting criteria, stopping criteria etc., depending on the business use case. Secondly and more importantly, it addresses the problem of cold start in E-commerce, wherein, new products need to be tagged to relevant MSKUs entirely based on their attributes.

As can be seen in Fig. 2(a-b), other multimodal algorithms, namely SSAHC and I-Louvain have a higher generalization error compared to DMDTC, on the attribute space. For I-Louvain, while the train (A+C) score remains high, the generalization on the attribute space is fairly low. On the other hand, K-Means, when implemented with a single modality of information (product attributes alone) performs worst, outlining the significance of cobrowse information in better exploiting the attribute space and generalizing the approach to freshly onboarded products. In contrast, DMDTC yields a consistently low generalization error across varying number of clusters (5-20), implying its robustness.

Experiments run on synthetic data for validation offered similar insights. For example, in Fig.3, I-Louvain achieves the best ARI score when the weight of pairwise-linkage (cobrowse) score is very high. The learning on attribute space, therefore, is limited as can be inferred from an inferior Gen. ARI value in Table 3 as well. Thus by identifying important attributes from a large attribute space and defining clusters on the shortlisted attributes, DMDTC can provide more tangible and actionable business insights. In fact, for many business use cases, product clustering output is useful only when the clusters can be defined based on attributes. Thus it makes sense to evaluate clustering performance on the attribute space, for it gives us a better sense of cluster output quality.

We performed an additional experiment on synthetic dataset to further check the robustness of our frameowrk. The number of noisy attributes in our dataset was increased from 5 to 15, without tweaking the continuous or categorical attributes. It could be seen (Table 3) that while the performance of DMDTC remains unchanged, that of I-Louvain and SSAHC drops significantly in both (A+C) and A space. Evidently, this is due to the nature of DMDTC algorithm, i.e. the irrelevant or noisy attributes are not considered for node splits and are consequently discarded. SSAHC, on the other hand, witnesses a maximum drop in performance because it utilizes all attributes for merging products into respective clusters and is not able to ignore the noisy or non-informative attributes.

To add to the discussion, since our algorithm groups products having similar cobrowse patterns into a single MSKU, cannibalizing products become easily identifiable. The generated MSKUs can also be utilized to understand and forecast demand patterns for various customer cohorts, eg. Gender, Geography, Affluence etc. in order to enable better product selection and user personalization.

4 CONCLUSION

The proposed methodology enables efficient integration of SKU attributes with products' co-browse information while ensuring interpretability, customer-centricity and scalability. The major findings of the study are highlighted below:

(1) The resultant MSKUs are more robust, interpretable, user-centric and address the cold start problem of product assignment, i.e. based entirely on attribute information.
(2) DMDTC outperforms SSAHC and I-Louvain on the Women Ethnic and Backpack datasets with significantly better Gen. SIL scores. On the synthetic dataset as well, the ARI of DMDTC on the (A+C) space is higher (0.81) than SSAHC (0.58) and comparable to I-Louvain, while the ARI on A space (0.78) is highest for DMDTC.
(3) Easily available co-browse and product attribute information makes the framework scalable across verticals. Further, it is easy to incorporate business guardrails in DMDTC making the resultant MSKUs useful in downstream tasks like designing pricing and selection strategies.

In a future study, we aim to investigate the relevance of these MSKUs for various customer cohorts to identify the right granularity of cohort selection and thus enable better user personalization.

ACKNOWLEDGEMENT

The authors wish to thank Dr. Saswata Sahoo (Senior Applied Scientist, Analytics, Flipkart) for valuable discussions.

REFERENCES

[1] Eric Bair1. 2013. Semi-supervised clustering methods. (2013). https://repository.library.northeastern.edu/files/neu:cj82sw041/fulltext.pdf

[2] Ozan Cakir and Mustafa S. Canbolat. 2008. A Web-Based Decision Support System for Multi-Criteria Inventory Classification Using Fuzzy AHP Methodology. 35, 3 (2008). https://doi.org/10.1016/j.eswa.2007.08.041

[3] Junxiang Chen. 2018. INTERPRETABLE CLUSTERING METHODS. (2018), 8–9. https://arxiv.org/abs/1307.0252v1

[4] Ching-Wu Chu, Gin-Shuh Liang, and Chien-Tseng Liao. 2008. Controlling inventory by combining ABC analysis and fuzzy classification. *Computers Industrial Engineering* 55, 4 (2008), 841–851. https://doi.org/10.1016/j.cie.2008.03.006

[5] David Combe, Christine Largeron, Mathias Géry, and Előd Egyed-Zsigmond. 2015. I-louvain: An attributed graph clustering method. In *International Symposium on Intelligent Data Analysis*. Springer, 181–192.

[6] Yukihiro Hamasuna, Yasunori Endo, and
and Sadaaki Miyamoto. 2012. On Agglomerative Hierarchical Clustering Using Clusterwise Tolerance Based Pairwise Constraints. *Journal of Advanced Computational Intelligence and Intelligent Informatics* 16, 1 (2012), 174–179. https://doi.org/10.20965/jaciii.2012.p0174

[7] Vladimír Holý, Ondřej Sokol, and Michal Černý. 2017. Clustering retail products based on customer behaviour. *Applied Soft Computing* 60 (2017), 752–762. https://doi.org/10.1016/j.asoc.2017.02.004

[8] Lawrence Hubert and Phipps Arabie. 1985. Comparing partitions. *Journal of classification* 2, 1 (1985), 193–218.

[9] TJ Van Kampen, R Akkerman, and DD Van Pieter. 2012. Multiple Criteria ABC Analysis with FCM Clustering. *International Journal of Operations Production Management* 32,7 (2012).

[10] Aydin Keskin and Coskun Ozkan. 2013. Multiple Criteria ABC Analysis with FCM Clustering. *Journal of Industrial Engineering* (2013).

[11] M Kumar and N R Patel. 2010. Using clustering to improve sales forecasts in retail merchandising. *Ann Oper Res* 174 (2010), 33–46.

[12] Shrenik Lad and Devi Parikh. 2014. Interactively Guiding Semi-Supervised Clustering via Attribute-Based Explanations. In *Computer Vision – ECCV 2014*, David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars (Eds.). Springer International Publishing, Cham, 333–349.

[13] W. James Murdoch, Chandan Singh, Karl Kumbier, Reza Abbasi-Asl, and Bin Yu. 2019. Interpretable machine learning: definitions,methods, and applications. (2019). https://arxiv.org/pdf/1901.04592.pdf

[14] Fariborz Y Partovi and Murugan Anandarajan. 2002. Classifying inventory using an artificial neural network approach. *Computers Industrial Engineering* 41, 4 (2002), 389–404. https://doi.org/10.1016/S0360-8352(01)00064-X

[15] Ramakrishnan Ramanathan. 2006. ABC inventory classification with multiple-criteria using weighted linear optimization. *Computers Operations Research* 33, 3 (2006), 695–700. https://doi.org/10.1016/j.cor.2004.07.014

[16] William M Rand. 1971. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association* 66, 336 (1971), 846–850.

[17] Meshal Shutaywi and Nezamoddin N. Kachouie. 2021. Silhouette Analysis for Performance Evaluation in Machine Learning with Applications to Clustering. (2021). https://www.mdpi.com/1099-4300/23/6/759/html

[18] Sudhir Voleti, Praveen K. Kopalle, and Pulak Ghosh. 2015. An Interproduct Competition Model Incorporating Branding Hierarchy and Product Similarities Using Store-Level Data. *Management Science* 61, 11 (2015), 2720–2738. http://www.jstor.org/stable/24551555