Leveraging customer shopping pattern to quantify and reduce ambiguity in Amazon's catalog

Subhrangshu Nandi, PhD
subhrn@amazon.edu

Wisam Hussain
wisah@amazon.edu

Andrew O'Connor
andreo@amazon.edu

Michelle Weis
weism@amazon.edu

Vishal Rathi
vrathi@amazon.edu

Christine Lee
chril@amazon.edu

Durga Ravichandran
durga@amazon.edu

Emilio Maldonado
emilim@amazon.edu

Abstract

Browse nodes are fundamental constructs of Amazon's hierarchical catalog where products are categorized based on product ontology and textual descriptions. Multiple Browse nodes might represent the same product concept duplicated throughout the catalog to accommodate for target audiences (Women, Men, Baby, and Pets), and customer segments (Consumer, Commercial). This duplication makes the nodes appear ambiguous to customers as they are unable to distinguish between them, for example, fitness trackers in sporting goods versus heart rate monitors in health department. This impacts product discovery, leads to frustrating shopping experience and erodes trust in Amazon. There is no mechanism that quantifies ambiguity between nodes in the catalog. The size of the catalog (28K nodes in the US marketplace), same products having multiple usages (example tote bags used as shopping bags and handbags), merchandising of combo products (example keyboard-mouse combos), are a few reasons why this is an intractable problem at Amazon's scale. In this paper, we use customer shopping pattern to significantly scale down the problem space to product concept levels and develop a mechanism to estimate ambiguity between node-pairs. The mechanism uses confusion matrix of gradient boosting decision tree based classifier on the product concept target label space. We apply the mechanism to 16 different product concepts, with 3,000 node-pairs, out of which 2% are marked as ambiguous by human specialists. Our mechanism achieves 84% precision and 70% recall in identifying the ambiguous node-pairs. This is the first data-driven approach that incorporates customer feedback to reduce ambiguity in Amazon's catalog. In three pilot projects, we successfully reduce up to 40% ambiguity in three node-pairs.

1 Introduction

When customers shop on Amazon website, products (ASINs) are returned in the search results and on the left navigation panel a list of destinations are displayed. These destinations are Browse nodes defined in Amazon's taxonomy. Browse enables customers' discovery experience by organizing Amazon's product selection into a Discovery Taxonomy for each Marketplace. The Taxonomy contains nodes representing top-level categories ("Electronics"), Product Types ("Televisions"), Assortments ("Small Appliances"), Refinements ("Screen Size"), and Merchandising nodes ("Camping Store"). Browse nodes are destinations where products are classified based on product ontology and textual descriptions. Sometimes, multiple Browse nodes might represent the same product concept duplicated throughout the Taxonomy to accommodate for target audiences (Women, Men, Baby, and Pets), and customer segments (Consumer, Commercial). For example, 35 browse nodes across 29 different departments contain women's leggings ASINs. In such cases, the list of browse nodes

7th Amazon Machine Learning Conference (AMLC 2019), Seattle, USA.

in search results is long. Consequently, it is hard for customers to distinguish between the product concepts represented in the different browse nodes, for example women's athletic pants (in fashion department) versus women's sports tights (sporting goods department). Our initial estimates suggest that there may be more than nineteen thousand [1] such node-pairs where product concepts may be hard to distinguish. In this paper, we describe a method by which, for the first time, we build a mechanism which takes feedback from customer shopping behavior, quantifies ambiguity between node-pairs in Amazon's catalog and recommends actions to reduce ambiguity.

1.1 Customer problem

If customers are offered similar products in different browse nodes, it is harder for them to distinguish between them. This was also highlighted in a 2018 user experience study on search customer frustrations [2]: *"Customers are frustrated and unclear which browse node to explore when there are multiple similar nodes or when they do not resonate with the naming of specific nodes"*. Figure 1 and table 1 list a few examples of such nodes in the catalog where customers might experience similar frustration.

Figure 1: Screenshot of keyword search and browse node navigation experience for "women's leggings"

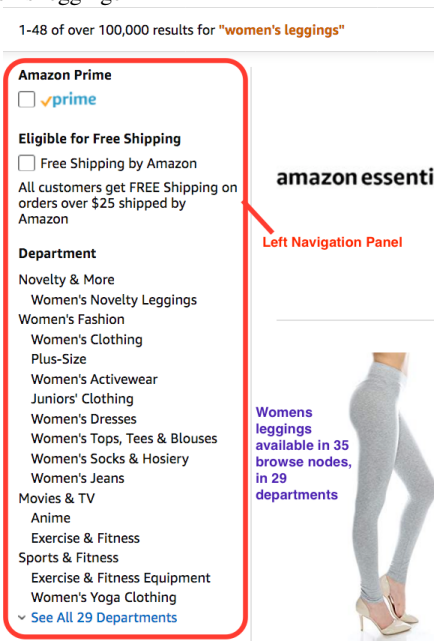


Table 1: Some examples of node-pairs representing very similar concepts in the US marketplace

concept	node 1	node 2
women's leggings	1046600 [us-fashion: women's athletic pants]	11444132011 [us-sporting-goods: women's sports tights]
women's leggings	1046600 [us-fashion: women's athletic pants]	9590811011 [us-sporting-goods: women's yoga leggings]
women's handbags	3421075011 [us-fashion: women's shoulder handbags]	2475899011 [us-fashion: women's cross-body handbags]
store signs	2896433011 [us-office-products: business & store signs]	490705011 [us-office-products: sign kits & poster kits]
activity trackers	5393958011 [us-sporting-goods: fitness trackers]	8619069011 [us-health: heart rate monitors]

1.2 Why this is a hard problem

The ever expanding Amazon catalog makes it a daunting task to ensure mutual exclusivity of all nodes. The worldwide Amazon selection contains over 12 billion non-media ASINs organized into marketplace-specific taxonomies. In the US marketplace, there are approximately 28 thousand nodes. Browse ontologists and taxonomists, subject matter experts of different product spaces, use competitor analysis, contextual awareness and language understanding to create browse nodes for different product concepts. However, due to reasons listed above it is not always possible to ensure mutual exclusivity between nodes. Once ambiguity is introduced into the taxonomy, it perpetuates. Figure 2 illustrates how ambiguity is perpetuated in the taxonomy. First, the sellers cannot distinguish

[1] see sec 6 for an explanation of this estimate

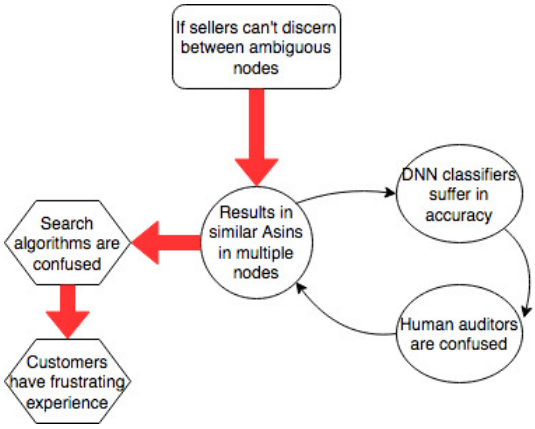
[2] See report 5 of 2018 user experience study on search customer frustrations

between browse nodes and either place the same products in multiple nodes, or in incorrect nodes. Second, human classification specialists are unable to decide on the best node for the product either. This results in poor quality labeled training data for ML classifiers. Third, ML-based classification systems in Browse for Auto-Classification (C@C, see Dubrov *et al.* (2018)) and Misclassification Detection (MDS, , see Ansari *et al.* (2018)) struggle because they train on the imprecise labeled data. Consequently, search algorithms show a long list of nodes representing the same product concept and customers have a frustrating experience.

1.3 Our solution to the problem

We start with the customer problem described in sections 1.1 and 1.2 and work backwards. Starting with a product concept c and its node(s) we identify which other nodes customers end up purchasing from. This is the customer discovery set N_c. Next, we obtain the most viewed ASINs in N_c and build a multi-class classifier with nodes in N_c as the classes. We estimate a classifier confusion matrix ("M_c", see Ting (2017)) and using methods similar to **?** and Liu *et al.* (2014) the ambiguity between nodes (classifier classes) after accounting for imprecise labels in the data. We name the system *ambiguous concepts intelligent detection* ("ACID"). To validate ACID output, we quantify pairwise ambiguity between nodes from 16 different concepts $c_1, \ldots, c_{20}$, spanning softlines, hardlines and consumables[3]. We audit ACID output with Browse taxonomists and get a precision of 84%, recall of 70% in identifying ambiguous node-pairs. ACID correctly identifies 99% of the node-pairs that are not deemed as ambiguous by taxonomists. Next, we conduct two pilot projects on products identified as ambiguous by ACID. In the pilot projects, Browse taxonomists use the same heuristics/methodology as they do for conventional (non-ACID) refreshes and merge the selection into a single node, removing the other from the customer facing taxonomy. We demonstrate that by merging nodes, we reduce ambiguity in the product space and improve customer experience in terms of better distinguishability between nodes in the concepts.

Figure 2: Perpetuating problem of ambiguity in Amazon's catalog eventually results in poor customer experience



This work is novel because for the first time we are able to learn from customer behavior, reduce the intractable problem space from 392 million catalog node-pairs [4] to node-pairs in customer discovery set and build a framework for automated computation of pairwise ambiguity.

The rest of the paper is organized as follows: Section 2 discusses related work, section 2.1 formulates the problem, section 3 describes the methods, section 4 highlights the important results. Section 5 summarizes the applications and customer impact and sections 6 and 7 concludes with a discussion on ongoing and future work.

2 Related work

A simple approach for quantifying ambiguity between two nodes is counting the proportion of ASINs that are assigned to both. While this is a good start, this approach is limiting, because multi assignment is only one of the many reasons causing ambiguity. For example, shoulder bag ASINs are very similar to cross-body bag ones and seem ambiguous to sellers and customers (see table 1). As a result, we end up with shoulder bag ASINs in cross-body bag node and vice versa. This creates ambiguity despite having different ASINs in the two nodes.

Another approach for quantifying ambiguity between nodes is estimating similarity between ASINs assigned to the nodes. Corpus-based and knowledge-based semantic similarity between text has been well explored in the literature (see Mihalcea *et al.* (2006), Kenter & De Rijke (2015) and many more on this topic). The idea is to represent the ASINs using trained embeddings and estimate similarity in the embeddings vector space. However, building such models require a large quantity of high quality

[3] see the Amazon wiki on product lines: https://w.amazon.com/index.php/ProductLine

[4] $\binom{28000}{2} = 391,986,000$

labeled training data that is representative of the rest of the catalog. ASIN2Vec (see Aggarwal *et al.* (2018)) is one such method which produces ASIN embeddings trained on the catalog data. While ASIN2Vec promises to produce representations learned from Browse taxonomy and ASIN texts, it has its limitations because the catalog assignment data is not clean and labeled training data is sparse and not representative of the rest of the catalog. C@C trains its own ASIN representations (Dubrov *et al.* (2018)). Although it has very high precision ($> 90\%$ in most categories of top 11 MPs) it suffers from low node coverage and is unusable for smaller nodes.

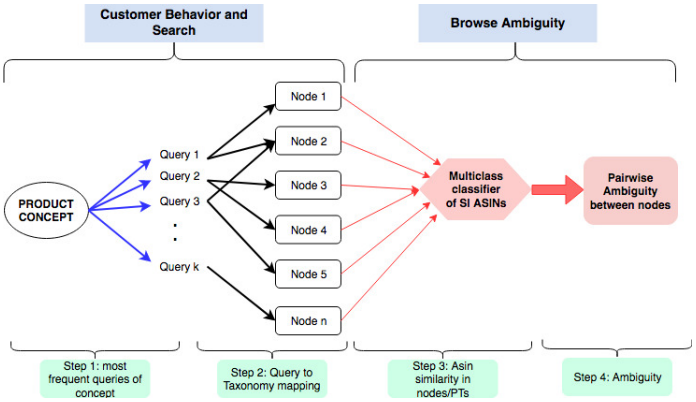
Regardless of which embeddings are used for representation, finding similarity between ASINs in the scale of Amazon's catalog is a daunting task. Nodes can be represented using an average of the ASINs assigned to them. Then, one could estimate a high dimensional correlation matrix between the nodes and identify node-pairs with high correlation as ambiguous. One could also employ matching algorithms such as kNN to quantify similarity between nodes. However, curse of high dimensionality (Keogh & Mueen (2010), Kouiroukidis & Evangelidis (2011), Radovanović *et al.* (2009) and many others discuss this issue) is a limiting factor in both the approaches.

To summarize, we need a method that is robust to imprecise training data, curse of dimensionality and that can scale to Amazon's catalog.

2.1 Problem formulation

Quantifying ambiguity in the catalog can become intractable if tackled at the marketplace level. With 28K nodes in the US marketplace it is computationally too expensive to quantify ambiguity in 392 million node-pairs. To tackle the problem at a more we leverage customers' shopping pattern. We focus on one product concept at a time. Figure 3 illustrates the mental model for our approach. *Step 1*: For a product concept c (examples of product concepts are in column 1 of table 1), we find the top customer search queries that lead to purchases, $Q_c = \{q_1, q_2, \ldots, q_k\}$. *Step 2:* For each $q_i \in Q_c$, we find all the browse nodes customers end up purchasing from by using *query browse affinity* (QBA) score [5]. We consider the union of purchasing browse nodes $N_c = \{bn_1 \cup bn_2 \cup \cdots \cup bn_n\}$ as the candidate set for quantifying ambiguity for concept c. We use customer behavior to obtain N_c, thus reducing the problem space dimension from 28K to cardinality of N_c.

Figure 3: Mental model of using customer behavior signals to quantify taxonomy ambiguity



Step 3: We obtain the most search impressed (SI) (most viewed by customers) ASINs in every node in N_c, represent them using an average of GloVe representation (see Pennington *et al.* (2014)) of their titles, bullet points and product descriptions and brands, and build a multi-class classifier on nodes in N_c as the classes. We choose SI ASINs instead of all ASINs in the catalog for two reasons: (1) the SI weighted catalog assignment accuracy is higher than the unweighted accuracy (2) search defect rate for top 8 ASINs for top 5K queries (on an average) is less than 6% (Goutam *et al.* (2017)) which is smaller than the rest of the ASINs. Hence, more viewed ASINs of a query have a higher chance of being relevant to the query. We choose gradient boosted decision tree (GBDT) based classifier (G_c) to build a multi-class classifier. In the first iteration, we estimate the confusion matrix in presence of imprecise labels using methods similar to Deng *et al.* (2016) and Liu *et al.* (2014). To tackle the imprecise labels in the data, we leverage the approach similar to computing alpha-trimmed central measures in high dimensional statistics and signal processing literature (see Chakraborty & Chaudhuri (2014), Febrero-Bande *et al.* (2012), Bednar & Watt (1984) to cite a few). In this approach, we drop the ASINs in the bottom $\alpha\%$ of prediction scores and retrain the classifier G_c^{α}.

[5] QBA scores is maintained by Search query understanding and used by Search relevance and matching algorithms for selecting ASINs to the the query search results (see Lin (2018), Yang *et al.* (2017))

Step 4: We compute the confusion matrix of G_c^α and obtain the pairwise ambiguity scores from the matrix.

For getting the top queries of a concept (step 1 in figure 3) we used search logs in TOMMY ASIN data [6] for a period of 3 months from August 2018 to October 2018. For getting nodes from queries (step 2) we used QBA[7] builds from October 2018. In step 3, to get the top SI ASINs we used Tommy data to compute SI and catalog data to get the latest assignments.

3 ACID methods

As explained in sec 2.1 ACID has two steps: first building a classifier, second computing the confusion matrix and estimating the pairwise ambiguity between nodes.

α-trimmed GBDT classification

To classify the set of SI ASINs S_c of concept c to nodes (target labels) in N_c, we use GBDT classifier using xgboost package v0.81 (Chen & Guestrin (2016)) in python 2.7.

1. Represent the ASINs using average GloVe embeddings of titles, bullet points, descriptions and brands.
2. Tune the GBDT hyper-parameters using 10-fold cross validation and AUC as the performance evaluation criterion.
3. Fit the classifier using the tuned parameters on the SI ASIN set S_c and compute the prediction score p_{ij} of each ASIN $s_i \in S_c$ for each class $bn_j \in N_c$.
4. Foreach ASIN $s_i \in S_c$, obtain the predicted class from $bn^* = \arg\max_j p_{ij}$. Denote $p_i = \max_j p_{ij}$
5. Compute the threshold p_α for α-trimming, such that $\mathbb{P}(p_i < p_\alpha) \leq \alpha$
6. Foreach $s_i \in S_c$ drop s_i if $p_i < p_\alpha$. Denote the reduced set as S_c'.
7. Refit the classifier G_c^α with ASINs in S_c' on the same classes in N_c.

Denote the accuracy of the classifier G_c^α as A_c^α and define ambiguity of product concept c as in eqn 1

$$\Lambda_c^\alpha = 1 - A_c^\alpha \tag{1}$$

If $\alpha <$ assignment defect rate of concept c then $A_c^\alpha \geq A_c$ and hence, $\Lambda_c^\alpha \leq \Lambda_c$. This way α-trimmed ambiguity will have reduced influence from imprecise labels arising from assignment defects. We can choose α to be fixed value or send out a manual audit to estimate the defect rate of nodes in concept c and set α to it. Here we set $\alpha = 0.05$. Next, we describe the computation of ambiguity from the classifier G_c^α.

Estimating confusion matrix and pairwise ambiguity

The confusion matrix M_c is estimated from the classifier G_c^α. In the confusion matrix shown in eqn 2 x_{ij} is the number of samples belonging to node bn_i but predicted to be in node bn_j. Total samples in node bn_i is simply $x_{i\cdot} = \sum_j x_{ij}$.

$$\begin{array}{c} \\ bn_1 \\ bn_2 \\ \vdots \\ bn_n \end{array} \begin{array}{c} \begin{array}{cccc} bn_1 & bn_2 & \cdots & bn_n \end{array} \\ \begin{pmatrix} x_{11} & x_{21} & \dots & x_{n1} \\ x_{12} & x_{22} & \dots & x_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ x_{1n} & x_{2n} & \dots & x_{nn} \end{pmatrix} \end{array} \tag{2}$$

Pairwise ambiguity between nodes n_p and n_q is defined as

$$\lambda_{pq} = \frac{x_{pq} + x_{qp}}{x_{p\cdot} + x_{q\cdot}} \tag{3}$$

Notice that $0 \leq \lambda_{pq} \leq 1, \forall p, q$. Higher values of λ_{pq} denote higher ambiguity between nodes n_p and n_q and vice-versa. Pairwise ambiguity scores have a linear relationship with the overall ambiguity of the concept, i.e., Λ_c is a weighed average of the pairwise ambiguity scores, the weights being the relative sizes of the nodes (see eq 4)

$$\Lambda_c = \sum_p \sum_{q>p} \lambda_{pq} \cdot w_{pq}, \quad \text{where } w_{pq} = \frac{x_{p\cdot} + x_{q\cdot}}{\sum_i x_{i\cdot}} \tag{4}$$

[6] https://w.amazon.com/index.php/Search/Analytics/Datasets/TommyASIN

[7] https://w.amazon.com/index.php/Search/A9/QueryUnderstanding/QBA2%20Precompute%20Build

Quantifying customer impact

Actionable insights from ACID include merging ambiguous nodes or deprecating one of them and moving ASINs to the other one. In either case, the taxonomy is updated. To test the impact of changing the taxonomy on customer experience, we need to do a weblab where customers are segmented into two different groups, with each group viewing ASINs from a different state of the taxonomy. However, there is no such functionality as Search cannot index two different taxonomies. So, we have to use offline measures to quantify the impact of taxonomy changes on customer experience. We use QBA scores as our measure of impact.

QBA scores are used by multiple teams to quantify how exhaustive a browse node is, for a search query. The QBA score $p_{q,n} \in [0, 1]$ for a search query q to a node n, is computed from past customer behavior and higher number means a higher proportion of the relevant ASINs of q are in node n. When nodes have distinguishable products, their queries have higher QBA scores. When multiple nodes $n_1, \ldots n_k$ have similar products, their queries have low QBA scores $p_{q,n_1}, \ldots, p_{q,n_k}$ to all of them, implying customers are unable to distinguish between the product concepts represented by nodes. For example, QBA scores of "women's leggings" query for the different nodes shown in figure 1 are shown in table 2. To quantify impact of taxonomy changes motivated by ACID, we compute QBA scores of 10 most frequent queries of the nodes before merging, and their scores to the merged node after merging. We do a paired t-test to obtain statistical significance. We use the top 10 queries mapped to a node similar to the ones that are merged, as a control group, to verify that the effect on QBA scores of the merged nodes was as a result of reducing ambiguity.

Table 2: QBA score of women's leggings query

node id	node name	qba score
11444119011	Women's Sports Clothing	0.10
9590811011	Women's Yoga Leggings	0.10
11444071011	Sports & Fitness Clothing	0.12
3456051	Women's Activewear	0.13
2419370011	Women's Yoga Clothing	0.13
2371064011	Yoga Clothing	0.13
3422251	Yoga Equipment	0.14
3407731	Exercise & Fitness Equipment	0.16
1258967011	Women's Leggings	0.51
1040660	Women's Clothing	0.71

4 ACID results: Evaluation and Validation

To evaluate ACID, we chose 16 different product concepts spanning across softlines, hardlines and consumables. We built G_c^α for each concept and computed pairwise λ_{pq} for each node-pair in the 16 concepts. Browse taxonomists independently evaluated the node-pairs to judge if they represented ambiguous concepts. Results are summarized in table 3.

There were a total $3,060$ node-pairs out of which ACID identified $53(1.73\%)$ as ambiguous (where $\lambda_{pq} > \lambda_{\text{threshold}}$)[8] and the taxonomists identified $63(2.06\%)$. Of the 53 node-pairs, taxonomists agreed with 44 of them being ambiguous either due to a taxonomy problem or a classification problem. The precision of ACID was $\frac{44}{53} = 83\%$, recall $\frac{44}{63} = 70\%$. ACID agreed with 99% of the node-pairs deemed unambiguous by the taxonomists. and an overall accuracy of 99%.

5 Applications and customer impact

ACID currently has the following use-cases: (1) detecting node-pairs that represent similar product concepts and quantifying the ambiguity at the concept level and the node-pair level (2) helping taxonomists align and define nodes under a 'global taxonomy' structure that is reused across marketplaces (3) ensuring when new nodes are created, ambiguity is not introduced in the taxonomy. Three pilot projects were selected based on ACID output as part of end-to-end integration plan.

5.1 Pilot projects identified by ACID

1. Travel mugs and tumblers: When customers shopped for travel mugs and tumblers they ended up purchasing from ten different nodes in the home and kitchen category. The the product space ambiguity was $\Lambda_c = 12.88\%$. The top three node-pairs with $\lambda_{pq} > \lambda_{\text{threshold}}$ were due to parent-child classification coverage problem, with many ASINs in parent node 13217501 being similar to its children nodes. This type of ambiguity can only be reduced by reassigning ASINS, which was out of scope for this experiment. Instead, we selected a node pair with taxonomy ambiguity: commuters & travel mugs (60208) and travel insulated drink tumblers (9630571011),

[8] We chose $\lambda_{\text{threshold}} = 0.05$

Table 3: Summarizing the results of ACID experiments in 16 product concepts in the US marketplace

concept	nodes	node-pairs	ambiguous ACID	ambiguous taxonomist	unambiguous ACID	unambiguous taxonomist	Λ_c	Λ_c^α
shoelaces	3	3	0	1	3	2	4.26%	3.10%
ties	5	10	0	3	10	7	0.53%	0.19%
shampoo	6	15	0	6	15	9	11.56%	10.07%
body pillow	7	21	0	6	21	15	3.84%	3.05%
water bottles	9	36	0	1	36	35	0.72%	0.00%
umbrella	12	66	5	3	61	63	12.48%	11.87%
baby toothbrush	14	91	2	2	103	103	26.73%	26.14%
stickers	17	136	2	2	134	134	9.94%	8.98%
smart bracelets	19	171	9	5	162	166	10.96%	22.14%
wasabi peas	19	171	3	2	168	169	8.60%	7.71%
fresh cut flowers	20	190	3	3	187	187	1.75%	1.10%
scissors	20	190	0	3	190	187	5.79%	4.95%
wheelbarrows	24	276	2	1	274	275	3.98%	2.71%
women's leggings	33	528	15	15	513	513	32.1%	31.71%
lipstick	34	561	2	1	559	560	4.07%	3.07%
stationery	35	595	10	9	585	586	27.55%	27.20%

with $\lambda_{pq} = 2.51\%$. Taxonomists merged the two nodes and named it travel mugs & tumblers (602608), reducing Λ_c to 12.24%.

2. Changeable letter boards: $\lambda_{pq} = 40.73\%$ between Changeable Letter Boards (2896436011) and Business & Store Changeable Letter Boards (490769011) nodes. After ACID quantified the pairwise ambiguity, upon deep dive it was revealed that the two nodes had very similar ASINs. The concept ambiguity was $\Lambda_c = 3.75\%$. After merging the two nodes it reduced to $\Lambda_c = 1.96\%$
3. Store signs: $\lambda_{pq} = 28.1\%$ between Sign Kits & Poster Kits (2896433011) and Business & Store Signs (490705011) nodes. In this case also, the two nodes had very similar ASINs. This project is ongoing. After merging, we expect the concept ambiguity Λ_c to reduce from 3.75% to 2.71%. Store signs and changeable letter boards were part of the same product concept - store signs. ACID helped identify two node-pairs with very similar ASINs in them. In both cases one of the nodes was in US-office-products department and the other one was in US-home-garden department.

Table 4 summarizes the ambiguity reduction as a result of merging or deprecating nodes identified by ACID.

Table 4: Ambiguity reduction in pilot projects based on ACID output

product concept	λ_{pq} **before refresh**	Λ_c **before refresh**	Λ_c **after refresh**
travel mugs & tumblers	2.51%	12.9%	12.2%
changeable letter boards	40.7%	3.75%	1.96%
store-signs	28.1%	3.75%	2.71%

5.2 Customer impact

The merged node travel mugs & tumblers (see sec 5.1) was launched on 01/24/2019, allowing us six weeks of post launch data to quantify the customer impact of the node merge. We quantified the impact on customers who primarily use search queries to discover products. We obtained the top 10 most frequent queries of both the nodes before merging and computed their QBA scores

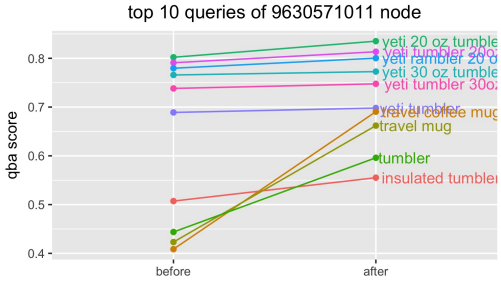
Table 5: QBA scores before and after merging travel mugs and tumblers, 13880501 is the control node

node id	before	after	Δ	p-value
9630571011	0.48	0.56	0.08 ($\uparrow$ 17%)	0.0004
602608	0.36	0.50	0.14 ($\uparrow$ 39%)	0.0005
13880501	0.62	0.55	0.06 ($\downarrow$ 7%)	0.9564

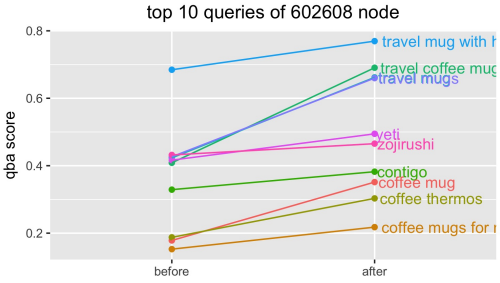
(p_q) before and after merging, then conducted paired t-test. Results are summarized in table 5. We also analyzed a control node (thermoses, node id 13880501, which is a sibling node of 602608). We see a 17% and 39% improvement in average QBA scores of the nodes that are merged, which indicates an improvement in confidence of distinguishability of products in the merged nodes. For the control nodes, the QBA scores of the top 10 queries either remained the same or decreased, indicating the increase in merged node queries was not common across similar nodes. We conclude that the improvement in the QBA scores of the queries mapped to the travel mugs and tumbler nodes is due to the merging. The p_q of top 10 queries of the two treatment nodes and the control nodes, computed before and after the merging, are shown in fig 4.

Figure 4: QBA scores of travel mugs and tumblers nodes, before and after the nodes were merged

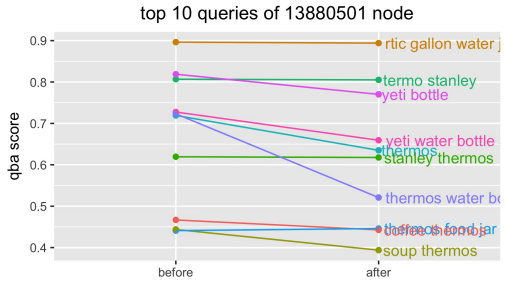
(a) p_q of the travel insulated tumblers node



(b) p_q of the commuters and travel mugs node



(c) p_q of the thermoses (CONTROL) node



6 Discussion and ongoing work

Of the 3,060 node-pairs representing 16 concepts in the experiment, taxonomists identified 63 (2.06%) ambiguous pairs. We estimate US marketplace to have 5K product concepts, and on an average 20 nodes per concept. ACID reduces the problem space from 392 million node-pairs to $\sum_{i=1}^{5k} \binom{20}{2} = \sum_{i=1}^{5k} 190 = 950$K node-pairs. Assuming 2% of them to be ambiguous, we expect to find 19K ambiguous node-pairs in Amazon's catalog. ACID is expected to play pivotal role in detecting them.

ACID has some limitations and need more research and experiments for improvement: (1) The ACID scores are not calibrated and may not be comparable across different categories. For example, a $\lambda_{pq} = 10\%$ in clothing may not have the same impact on customers as a $\lambda_{pq} = 10\%$ in electronics. (2) Choosing $\lambda_{\text{threshold}}$: In absence of prior work in this area $\lambda_{\text{threshold}}$ was chosen at 0.05. Instead, it could be chosen based on customer signals and potentially could be different for different categories. (3) Choosing parameter α: α-trimming reduces the impact of imprecise labels on ACID scores. More experiments are needed for selecting the right value. It depends on defect rates of the concepts and could be different for different concepts. Of the 16 concepts in only "smart bracelets" $\Lambda_c^\alpha > \Lambda_c$. Everywhere else, α-trimming helped reduce the effect of imprecise labels on the classifiers.

The novelty of ACID lies in reducing the problem space from the whole catalog to the product concept using customer signals. Using SI ASINs to build the classifier makes ACID scores more customer focused. α-trimming is another innovation that helps reduce the effect of imprecise labels on the classifier's output. Using classifier confusion matrix to quantify ambiguity instead of correlation matrix or other distance based methods circumvents the curse of dimensionality. In summary, using innovation at different stages, this paper presents a working solution to a complex and unsolved customer problem.

7 Conclusion

We started with the customer problem of not being able to distinguish between Browse nodes representing similar product concepts in Amazon's catalog. We used customer shopping pattern to scale down the problem space to product concepts. Next, we developed ACID to quantify ambiguity between node-pairs, using confusion matrix of an α-trimmed gradient boosted decision tree classifier. We validated the output of our method on 16 different concepts spanning three product lines in the US marketplace. Using ACID's recommendations we successfully identified ambiguity in three product concepts, performed necessary changes to the nodes and reduced ambiguity in all three of them.

References

Aggarwal, V, Volozin, A, Ansari, H, Chakraborty, S, & Sinha, A. 2018. ASIN2Vec: A taxonomy-based embedding for products. *Proceedings of the 6th Amazon Machine Learning Conference 2018*.

Ansari, M. Hidayath, Chakraborty, Subhadeep, Sinha, Avik, Mahendru, Ajay, & Sekine, Philippe. 2018. Identifying Miscategorized Products. *Proceedings of the 6th Amazon Machine Learning Conference 2018*.

Bednar, J, & Watt, T. 1984. Alpha-trimmed means and their relationship to median filters. *IEEE Transactions on acoustics, speech, and signal processing*, **32**(1), 145–153.

Chakraborty, Anirvan, & Chaudhuri, Probal. 2014. On data depth in infinite dimensional spaces. *Annals of the Institute of Statistical Mathematics*, **66**(2), 303–324.

Chen, Tianqi, & Guestrin, Carlos. 2016. Xgboost: A scalable tree boosting system. *Pages 785–794 of: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. ACM.

Deng, Xinyang, Liu, Qi, Deng, Yong, & Mahadevan, Sankaran. 2016. An improved method to construct basic probability assignment based on the confusion matrix for classification problem. *Information Sciences*, **340**, 250–261.

Dubrov, Bella, Joshi, Ashutosh, Bondre, Tejas, Jawa, Hitesh, & Sinha, Avik. 2018. Deep Learning for Text-Based Classification of Products into Browse Nodes. *Proceedings of the 6th Amazon Machine Learning Conference 2018*.

Febrero-Bande, Manuel, de la Fuente, M Oviedo, *et al.* . 2012. Statistical computing in functional data analysis: The R package fda. usc. *Journal of statistical Software*, **51**(4), 1–28.

Goutam, Rahul, Ho, Christopher, Headden, William, & Jammalamadaka, Ravi. 2017. Search Defect Aware Reranking. *Proceedings of the 5th Amazon Machine Learning Conference 2017*.

Kenter, Tom, & De Rijke, Maarten. 2015. Short text similarity with word embeddings. *Pages 1411–1420 of: Proceedings of the 24th ACM international on conference on information and knowledge management*. ACM.

Keogh, Eamonn, & Mueen, Abdullah. 2010. Curse of dimensionality. *Encyclopedia of machine learning*, 257–258.

Kouiroukidis, Nikolaos, & Evangelidis, Georgios. 2011. The effects of dimensionality curse in high dimensional knn search. *Pages 41–45 of: 2011 15th Panhellenic Conference on Informatics*. IEEE.

Lin, Heran. 2018. Hierarchical Multi-label Classification of Queries to Browse Categories. *Proceedings of the 6th Amazon Machine Learning Conference 2018*.

Liu, Zhun-Ga, Pan, Quan, & Dezert, Jean. 2014. A belief classification rule for imprecise data. *Applied intelligence*, **40**(2), 214–228.

Mihalcea, Rada, Corley, Courtney, Strapparava, Carlo, *et al.* . 2006. Corpus-based and knowledge-based measures of text semantic similarity. *Pages 775–780 of: AAAI*, vol. 6.

Pennington, Jeffrey, Socher, Richard, & Manning, Christopher. 2014. Glove: Global vectors for word representation. *Pages 1532–1543 of: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*.

Radovanović, Miloš, Nanopoulos, Alexandros, & Ivanović, Mirjana. 2009. Nearest neighbors in high-dimensional data: The emergence and influence of hubs. *Pages 865–872 of: Proceedings of the 26th Annual International Conference on Machine Learning*. ACM.

Ting, Kai Ming. 2017. Confusion matrix. *Encyclopedia of Machine Learning and Data Mining*, 260–260.

Yang, Chao, Tan, Bo, & Patnia, Abhishek. 2017. Mapping Query To Browse Nodes Using Deep Learning Methods. *Proceedings of the 5th Amazon Machine Learning Conference 2017*.