Islands of Interest: Mining Concentrations of User Search Intent over E-commerce Product Categories

Neeraj Pradhan, Vinay Deolalikar, Deependra Singh
Search, Data Mining, Machine Learning & Personalization
Groupon
Palo Alto, CA 94306

Abstract. E-commerce is burgeoning. Within the last few years, both in the US and the UK, E-commerce has overtaken brick and mortar stores for retail. E-commerce comes with its own set of core problems and challenges: constructing recommendations for users; designing product taxonomies so that inventories can be classified accurately, and in a way that the user finds easy to navigate to; facility allocation for inventory, so that shipping costs are minimized, to name a few. The first of these problems has received much research attention. Accordingly, there is abundant literature on building such recommendations using user's browsing data.

Let us now consider E-commerce *data*. Search as a component of E-commerce shows a steady rise, and has, in many cases, overtaken browse as the primary means of user interaction with E-commerce sites. Accordingly, E-commerce generates more search data than browse data. However, search data is inherently "local" to a query. Therefore, it is not immediately obvious whether it can be used to build "global" (i.e., where no query is involved) knowledge. On the other hand, each of the three problems mentioned earlier requires such global knowledge.

In this paper, we introduce a global structure—that we call *islands of interest*—that is mined from local search data. We show that islands of interest are highly relevant to each of the E-commerce problems mentioned earlier. We introduce two algorithms—one based on community detection, and the other on clustering—that can identify islands of interest. We build a framework that can compare the islands identified using these two approaches, as being efficacious in solving our motivating E-commerce problems. We believe that in addition to insights into user behavior, islands of interest will provide further impetus to work on hitherto lesser known problems of E-commerce such as design of product taxonomies and facility allocation for inventory.

1 Introduction

E-commerce is one of the lasting legacies of the internet era. In the past five years, e-commerce retail sales have overtaken brick and mortar in both the US and the UK[1].

E-commerce comes with its own set of technical challenges. Some have received extensive research attention, such as building recommendations for users. Other challenges have not hitherto received much attention, although they are core to e-commerce. We list below three challenges that motivate our work.

Recommendations. Recommendations are a core part of the consumer e-commerce experience. Recommendations are broadly of two types: cross-sell and up-sell. Broadly, cross-sell refers to a recommendation that tries to sell an item that "goes well with" the item the user is viewing or has chosen for purchase. For example, batteries, along with electronic toys. In contrast, up-sell is when the e-commerce site recommends an alternative or substitute item to the one the user has chosen. The techniques in this paper are applicable to both scenarios.

Inventory placement. E-commerce sites provide the user with two types of inventory. The first type is external, or third party inventory. An example of such inventory would be third party merchants on, say, Amazon. In addition, e-commerce majors also stock their own goods on a large scale. These goods are spread across godowns that are geographically located so as to be efficient for shipping times to various

[1] `http://tinyurl.com/zpozmsk`

markets. Further, the vendor has to consider which inventory should be co-located. This is done, for example, by considering co-purchases. Returning to the example of electronic toys and batteries: it makes eminent sense to co-locate these in the same godown to cut processing and shipping times and costs.

Taxonomy design. The *customer-facing taxonomy*[2], which we simply refer to as the *taxonomy*, is central to product layout in e-commerce. The taxonomy is usually of the form of a multi-level directed acyclic graph (Fig. 1) and mediates the user experience in two important ways—by directly providing a navigational channel to browse the products within each category, or indirectly through recommendations such as the "related items", and "you may also like" widgets seen on most e-commerce sites. In other words, several e-commerce sites show deals related to an item by ranking other items that are close to it in the taxonomy using a metric (such as popularity), and then showing a user the top ranked items. The first of these

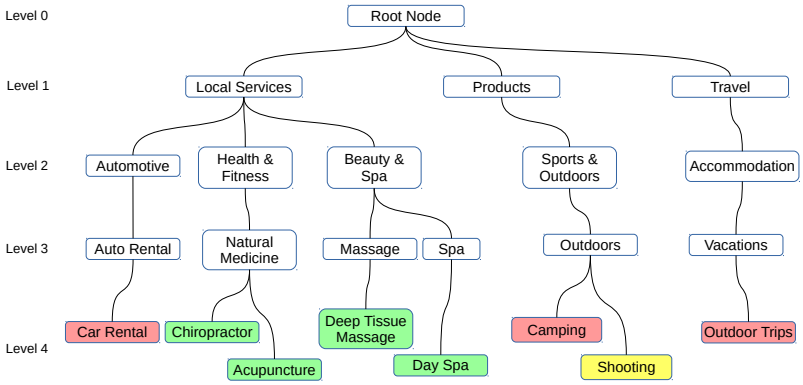

Fig. 1. An example of a Product Taxonomy showing color-coded concentration of user-intent across leaf nodes that are far apart in the taxonomy tree.

challenges has received much research attention, while the latter two have not.

Let us now commence a discussion about e-commerce *data*. Users interact with e-commerce sites primarily through two means: search and browse. Search involves specification of an intent, expressed through a query. The e-commerce site then responds by showing products that match that specific intent. Browse is typically more free-form, and a single browse session might incorporate diverse product views.

Let us discuss the relative proportions of search and browse. At our company CorpX—a major multi-billion dollar e-commerce vendor—search is the fastest growing component of traffic, and has crossed browse in overall traffic generation. We believe that similar statistics hold true at other e-commerce majors. At any rate, search is a highly significant component of e-commerce.

What does this mean for data mining? It means that search data is larger than browse data, and growing faster. Therefore, it makes eminent sense to mine this huge data source for building data mining algorithms to address e-commerce challenges.

However, Internally, some e-commerce companies may say that search is *local* and browse is more *global*. Since search is specific, and an intent has been specified at the start of the search, the conventional wisdom is that search is not very useful to construct algorithms that require a more global view. Each of the three challenges listed earlier require such a global view: none of them are specific to any notion of a search query. The local vs. global viewpoint says that search data may not provide us with the kind of insights that are needed for solving such global problems. Consequently, taking the example of recommenders, there is scant

[2] This is the taxonomy that the user can view, and navigate. Consequently, it is public. In addition, most e-commerce sites have an internal taxonomy that is often finer than the customer-facing taxonomy. In this paper, we will deal only with the customer-facing taxonomy.

literature on building recommenders using search data, and search data remains untapped for the purpose of building recommenders.

In this paper, we address this gap in the landscape of data mining applied to core e-commerce challenges. We build a framework that allows us to mine search data to address these challenges. Our framework combines an assortment of techniques from data mining—one-mode projections [27, 15], random walks, community detection [20], spherical k-means clustering[7]—and "patches together" local search data to yield global constructs that are immediately relevant to addressing the challenges listed earlier.

Our work is being productionized at CorpX.

We conclude this section with an outline of our contributions.

1. We show how to use massive, and fast growing search data to address core challenges in e-commerce. To our knowledge, the application of search data was hitherto limited to problems in retrieval.
2. We introduce problems such as taxonomy design using data mining considerations into the literature.
3. We introduce techniques such as one-mode projection and community detection to addressing these e-commerce challenges. Although these techniques have found some usage in e-commerce recommendation building, we expand the scope of their applications.

2 Background and Preliminaries

In this section, we introduce the customer-facing taxonomy and query-click graph in the context of e-commerce. We also define some quality measures for taxonomy design and distance measures for various taxonomy partitions with respect to the original tree.

2.1 Customer Facing Taxonomy

The customer-facing taxonomy is designed to provide an intuitive categorization for users to navigate and explore the products in an e-commerce application. Fig. 1 shows a subset of the product taxonomy. The *depth* of the taxonomy Δ is defined by the number of levels that exist below the root node in the taxonomy. In Fig. 1, the depth of the taxonomy is 4.

A *leaf node* in a taxonomy is one that has no children. Products and services are assigned to a single (and sometimes to a small number) leaf node. Distinct products assigned to the same leaf node are usually very similar. Products assigned to sister leaf nodes (i.e. leaf nodes that have the same parent node) are also similar, but somewhat less so. Let us denote our customer-facing taxonomy by $\mathcal{T}$, and the set of n leaf nodes by $\mathcal{C}=\{c_1,c_2,..,c_n\}$.

A *taxonomy partition* $\mathcal{P}$ is defined as the partitioning of $\mathcal{C}$ into K components, i.e. $\mathcal{P}=\{P_1,P_2..,P_K\}$, where $K\geq 1$ is referred to as the *size* of the partition. Each leaf node falls into exactly one component. The *implicit partition* of the taxonomy $\mathcal{T}$ is defined as the partitioning of $\mathcal{C}$ such that each component consists of all leaf nodes that are children of the same parent node in $\mathcal{T}$.

2.2 Query-Click Graph

Product search engines on e-commerce sites aim to match a user's intent with the inventory of products indexed in the database and rank them by their relevance to the given query. They are an important channel to users for both intent matching and product discovery.

The clicks generated for a given query term over the product inventory, when aggregated over all users, can be an important relevance signal. In this study, instead of looking at click flow from queries to individual products, we examine the flow to different leaf nodes of the customer-facing taxonomy. Aggregating the query-click flow over the leaf nodes also negates issues related to sparsity of click volume over individual products [26].

Let us denote the set of m queries in our model by $\mathcal{Q}=\{q_1,q_2,..,q_m\}$. We can connect a query q_i with a category c_j by an edge, whose weight $w_{i,j}$ is given by the click volume going from q_i to c_j. We represent the edge weights by the weight matrix $\mathbf{W}:\mathbb{R}^{m\times n}$, where $w_{i,j}$ is the (i,j) entry in $\mathbf{W}$. The *query signature* for the category c_i is defined as the click volume flowing from $\mathcal{Q}$ to c_i, and is denoted by the column vector $\mathbf{w}_i$,

$$\mathbf{W}=(\mathbf{w}_1,\mathbf{w}_2,..,\mathbf{w}_n).$$

For certain algorithms, we look at the transpose of the weight matrix, namely $\mathbf{W}^T$, where each row corresponds to a leaf category's query signature. Let us denote the resulting bipartite graph by $\mathcal{G}(\mathcal{V},\mathcal{E},\mathbf{W})$, with the vertices $\mathcal{V}=\mathcal{Q}\cup\mathcal{C}$ and edges $\mathcal{E}=\mathcal{Q}\times\mathcal{C}$.

3 Related Work

Our work touches upon many areas at the junction of ecommerce and data mining— recommender systems, community detection and clustering in ecommerce, information retrieval using query-click graph and the problem of taxonomy design and its utilities.

Recommender systems provide users with personalized suggestions from a large set of items. Traditionally, recommender systems have been clssified into *content-based filtering* and *collaborative filtering*(CF). *Content-based* recommendation systems recommend items similar to those a given user has liked in the past.[12] reviews the state-of-the-art as well as the most adopted content-based recommender systems. *Collaborative filtering* methods infer user's preferences from remaining users having similar tastes. In [23], Su and Khoshgoftaar present a survey of CF techniques and concisely deal with the inherent challenges such as sparsity and scalability etc. In practise, recommender systems that combine both of the techniques— called *hybrid recommender systems* [2]— are more prevalent. In our work, we mine search data to facilitate recommendations unlike existing techniques which utilize browse activities.

In ecommerce, community detection algorithms have been explored in order to assist CF. Using static community detection algorithms, Kamahara et al. [9] have proposed a community-based approach for recommender systems which can reveal unexpected users interests. Another aim behind incorporating community detection to recommendation is to provide a solution to the cold start problem, and this idea was proposed by Sahebi et al. [21]. More recently, [9] use dynamic community detection to incorporate the temporal aspect of the community structure in order to enhance the existing community based recommendations. In all the aforementioned works, community detection is applied to *browse* data in order to generate user or item communities, while we in this paper, perform community detection on *search* data. Similarly, clustering techniques have also been used to assist recommender systems in E-commerce. Sarwar et al. [22] address the issue of sparsity and scalability in CF by clustering users through k-means clustering, while O'Connor et al. [18] use clustering algorithms to partition the set of items based on user rating data to achieve the same objective. [24] use co-clustering techniques to simultaneously obtain user-item subgroups. in all the works mentioned, clustering techniques are applied to *browse* data and *search* data is left untouched.

The bipartite graph of query-click data has been well studied to provide query suggestions [14] and rank retrieved documents [6]. The authors in [14] use hitting time from random walks to rank queries, and in [6], the authors use forward and backward walks to rank images associated with a given query. Besides the bipartite graph, short random walks on derived graphs like the query-flow graph, a unipartite directed graph containing query transition probabilities, has also been studied for query suggestions [3]. The query-click data has also been used for agglomerative co-clustering of queries and web URLs [1]. Our motivation for doing a random walk on the query-click graph is to uncover the community structure of manually generated taxonomies that correspond to a concentration of user intent flow.

A fulfilment taxonomy is designed to be comprehensive, standards-driven, granular and designed to interface with the external environment [8]. The fulfilment taxonomy has been studied widely, with particular regard to the problem of mapping, integrating and updating products in such standardized eCommerce taxonomies [11, 19]. Click log data has also been used [16] for the problem of taxonomy matching. The customer-facing taxonomy in ecommerce is different from the backend standardized taxonomy used for fulfilment, and is designed to facilitate product discovery and ease of browsing for users. To the best of our knowledge, there is no existing literature on the design and study of user interaction with customer-facing taxonomies in ecommerce. For the most part, such taxonomies are hand-crafted based on human intuition, and evolve very slowly due to the technical costs involved. This is a recurring investment of time and money, and underscores the importance of understanding what constitutes a well-designed customer-facing taxonomy.

Importance of taxonomies is further accentuated by the fact that they are also used in recommender systems. In [5, 26, 10], the authors project user interest over a product taxonomy to provide recommendations through collaborative filtering using implicit user signals in the presence of information sparsity. Matos et al. in [13] propose various strategies to improve content-based recommender systems using term descriptors for taxonomy nodes. The underlying taxonomy space in these recommender systems is treated as fixed. The authors in [4] readjust the taxonomy to have a relatively even transaction distribution over the considered nodes for recommendations. There is, however, no structural change to the original taxonomy.

The communities from the taxonomy partition revealed through our work is likely to bring together leaf nodes which may be far apart in the original taxonomic tree with an aim to achieve high concentration of user intent and high similarity of within-community nodes.

4 Algorithms

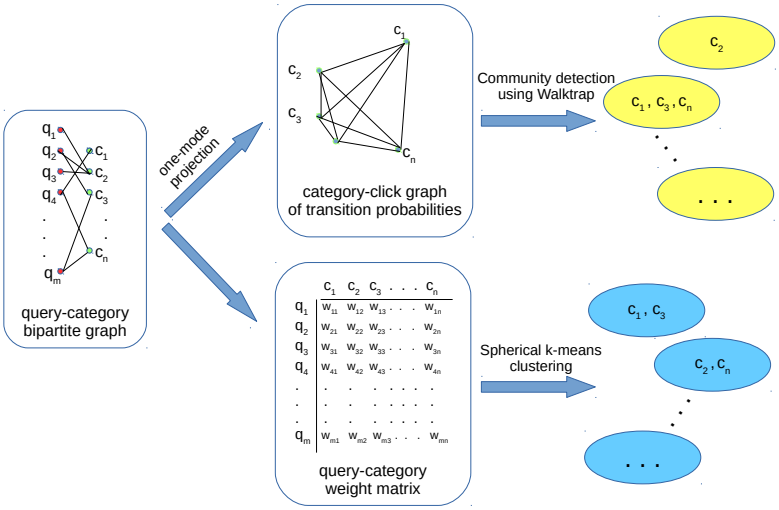

Fig. 2. Schematic of algorithm IslandComms (top) and algorithm IslandClusts (bottom)

4.1 The Algorithm IslandComms

One Mode Projection. Many real-world networks can be modeled as bipartite graphs where edges capture the interaction between two *modes* or distinct sets of entities (in this case, $\mathcal{Q}$ and $\mathcal{C}$) [17, 25]. In many cases, we are primarily interested in modeling the relationship amongst nodes in one of these modes. A common approach is to use *one-mode projection* [27, 15] which models edge weights between nodes in one of the modes by incorporating information about their connectivity in the original bipartite graph. The query-category graph, however, is weighted and there is scant literature on the problem of one-mode projection of a weighted bipartite graph.

Some advantages in analyzing the one-mode projection of the original graph is availability of many existing network analysis algorithms typically designed for one-mode graphs and, in some cases, the computational benefits of running these algorithms on the reduced size graph. In our case, the second benefit is particularly relevant as $|\mathcal{C}| \ll |\mathcal{Q}|$.

Projecting the query-category graph on to $\mathcal{C}$ results in a graph that connects different leaf nodes within $\mathcal{C}$. The projected weight matrix $\mathbf{W}^{\mathcal{C}} : \mathbb{R}^{n \times n}$ models the strength of the connection between the different leaves, with $w^{\mathcal{C}}_{i,j}$ denoting the edge weight between category c_i and category c_j in the projected graph. We shall refer to this projected graph as the *category-click graph*. We describe a projection technique to project the query-category graph on to $\mathcal{C}$, that we found to be effective in practice.

Cosine Similarity Projection. The cosine similarity projection models the strength of the connection between c_i and c_j by the cosine similarity of the click distribution coming from $\mathcal{Q}$ given by their query signatures, $sim_{cos} = \frac{\mathbf{w}_i \cdot \mathbf{w}_j}{\|\mathbf{w}_i\| \|\mathbf{w}_j\|}$. We apply a threshold θ to remove the long tail of noisy edges connecting dissimilar categories, so that the projected weight is given by

$$w^{\mathcal{C}}_{i,j} = \begin{cases} sim_{cos}(c_i, c_j) & \text{, if } sim_{cos}(c_i, c_j) \geq \theta \\ 0 & \text{, if } sim_{cos}(c_i, c_j) < \theta. \end{cases}$$

Pruning the large number of noisy edges also has a computational benefit as it leads to a sparser $\mathbf{W}^{\mathcal{C}}$. By using the cosine similarity projection technique, we obtain a much smaller projected weight matrix $\mathbf{W}^{\mathcal{C}}$ which is better suited for running community detection algorithms with high time complexity.

Random Walk. Projection of the query-category graph on to the set of categories $\mathcal{C}$ gives us a one-mode weighted graph with edges representing the strength of the association between two given categories. We would then like to partition this graph into communities so that the flow across communities is minimized while preserving locality within each of these communities. We can do random walks on this one-mode graph where the probability of transitioning from a given node to a neighboring node is proportional to its weight,

$$p_{i,j} = \frac{w^{\mathcal{C}}_{i,j}}{\sum_j w^{\mathcal{C}}_{i,j}}.$$

The underlying idea behind this approach is that the transition probability between two nodes can be treated as a proxy for the flow between them. Therefore, if we are able to find communities that have relatively higher edge weights between nodes of the community as against edges crossing the communities, the query click distribution over the induced taxonomy partition is likely to be concentrated within communities. We use one such community detection algorithm, *Walktrap*, on the projected graph that utilizes short random walks on the graph to find communities of like nodes.

Island detection using Walktrap communities. The idea behind the Walktrap algorithm is that short random walks starting at any given node tend to get trapped within communities [20]. After t random hops, the probability transition vector of a node gives the probability of reaching any other node from the given node. The probability transition vectors of two nodes that have dense inter-connections is likely to be similar. Walktrap defines a distance metric over the space of probability transitions and uses this metric to do a hierarchical agglomerative clustering.

Parameters in IslandComms. Note that there are two tuning parameters — the number of steps in the random walk t, and the threshold θ used in the cosine similarity projection (Sec. 4.1).

4.2 The Algorithm IslandClusts

For our baseline, we discover category communities directly from the query-click data over $\mathcal{C}$ by clustering the query signatures of different categories given by $\mathbf{W}^T$ (Sec. 2.2). Standard k-means with euclidean distance does not give good clustering results on the category-click data due to the high dimensionality of the query signatures as $|\mathcal{C}| \ll |\mathcal{Q}|$. As a solution, we can treat $\mathbf{W}^T$ as a document-term matrix, and use spherical k-means [7], an efficient document clustering algorithm that uses the cosine distance metric.

Parameters in IslandClusts. Note that there is only one tuning parameter — number of resultant islands K.

In the next section, we will be using the framework discussed above to identify islands of interest in the leaf nodes of the taxonomy, and study their properties.

5 Properties of Islands

In this section, we introduce four quantitative measures that allow us to characterize partitions of a taxonomy. The first two of these measures capture properties that might be deemed desirable in islands of interest. The second two allow us to quantify how different, in terms of location relative to the taxonomy, an island of interest is to the implicit partitions of the taxonomy.

Locality. We are ready to introduce the first of our four quantitative measures. A property that we would intuitively associate with an island of interest is that its constituent leaves are "similar." This is motivated by a similar implicit assumption made in the construction of taxonomies: namely, that items that are placed

close together in the taxonomy are likely to be "similar". The notion of similarity between two leaves c_i and c_j is defined in terms of the cosine similarity of their *query signatures*,

$$sim_{cos}(c_i, c_j) = \frac{\mathbf{w}_i \cdot \mathbf{w}_j}{\|\mathbf{w}_i\| \|\mathbf{w}_j\|}.$$

We want to define the notion of *locality* to capture the above property. Given a taxonomy $\mathcal{T}$ and its partition $\mathcal{P}$, we can define locality as the weighted average cosine similarity of the categories within the same community,

$$\lambda^{\mathcal{P}} = \frac{\sum_{k=1}^{K} \sum_{i,j \in P_k; j>i} sim_{cos}(c_i, c_j)}{\sum_{k=1}^{K} \binom{|P_k|}{2}}.$$

Note that high locality favors higher number of partitions K with each partition P_k containing leaf nodes with high similarity.

Intent Flow. In order to measure the concentration of user intent over any partition of the taxonomy, we define the notion of *flow of intent*, or simply *flow*. Note that high locality does not imply low flow, as we can have a partition with many components containing like nodes but, user intent, measured in terms of the distribution of query clicks over this partition, could be highly diffused as depicted in Fig. 3. We quantify this notion below.

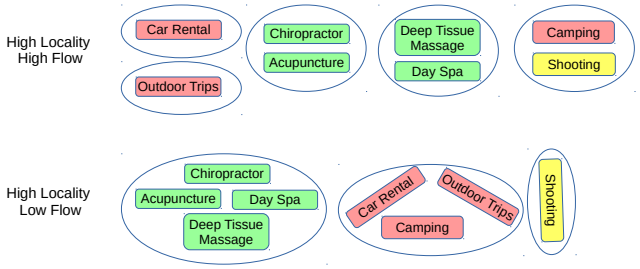

Fig. 3. High locality, high flow(top) and high locality, low-flow(bottom). Each color corresponds to a specific user intent. Top partition has islands with like nodes, but user intent, as measured in terms of the distribution of query clicks over islands, is relatively more diffused as compared to bottom partition.

We define the click volume from a query q_i to a specific $P_k \in P$ as the aggregated clicks to all leaves belonging to P_k,

$$V_{i,k} = \sum_{j \in P_k} w_{i,j}.$$

The click volume of a query q_i is defined as the aggregated clicks over all leaf nodes $\mathcal{C}$ for q_i,

$$V_i = \sum_{j=1}^{n} w_{i,j}.$$

Given a taxonomy $\mathcal{T}$, its partition $\mathcal{P}$, and a query q_i, the intent flow $\xi_i^{\mathcal{P}}$ for q_i is defined as the entropy of the click distribution of q_i over $\mathcal{P}$,

$$\xi_i^{\mathcal{P}} = \sum_{k=1}^{K} \frac{V_{i,k}}{V_i} \ln\left(\frac{V_{i,k}}{V_i}\right).$$

The flow of intent for a taxonomy partition $\mathcal{P}$ is defined as the aggregate flow of intent over all queries $q \in \mathcal{Q}$ weighted by the click volume for the query,

$$\xi^{\mathcal{P}} = \frac{\sum_{q=1}^{m} V_q \xi_q^{\mathcal{P}}}{\sum_{q=1}^{m} V_q} = \frac{\sum_{q=1}^{m} \sum_{k=1}^{K} V_{q,k} \ln\left(\frac{V_{q,k}}{V_q}\right)}{\sum_{q=1}^{m} V_q}.$$

Note that low flow tends to favor partitions with smaller size as the smaller size leads to a natural concentration of query click volume. One way to think of flow is in terms of separation of items in the

taxonomy which are closely related in intent. Therefore, while high locality is a measure of *precision*, low flow can be considered a *recall* measure.

While locality and flow give us a framework to compare the quality of partitions, we would further like to know how "different" these partitions are from the implicit partition. To develop a measure of distance of a partition from the customer-facing taxonomy, we introduce the notion of *girth* and *expansion coefficient* of a component of a partition.

Girth. Let us define the *taxonomic distance* $d(c_i,c_j)$ between two nodes c_i and c_j as the maximum of the height from the lowest common ancestor of c_i and c_j to either of c_i or c_j in the taxonomy tree $\mathcal{T}$. Note that for the given taxonomy (Fig. 1), $1 \leq d(c_i,c_j) \leq \Delta \; \forall c_i,c_j \in \mathcal{C}$. The taxonomic distance is 1 in the case that c_i and c_j are leaf sister nodes, whereas it is maximum (Δ) if c_i and c_j have the root as the lowest common ancestor. We then define the girth of a component $P_k \in P$ as the average taxonomic distance between any two nodes in P_k,

$$\rho(P_k) = \frac{\sum_{i,j \in P_k; j>i} d(c_i,c_j)}{\binom{|P_k|}{2}}.$$

Expansion Coefficient. Given a set of leaf nodes in a component P_k, we define the *closure* $cl(P_k)$ as the superset that is closed under the operation of taking sister nodes. The expansion coefficient of a component is then defined as the ratio of the size of its closure and the component's size,

$$\eta(P_k) = \frac{|cl(P_k)|}{|P_k|}.$$

As an example, if a component's expansion coefficient is 2, it implies that around half the leaf nodes from related components in the implicit partition were picked up to form this component. Note that the girth and expansion coefficient of all components in the implicit partition is one. For a partition $\mathcal{P}$, we define the *median girth* and *median expansion coefficient* as the median of the respective measures for all the components in $\mathcal{P}$.

6 Dataset and Protocol

Dataset. We conduct our empirical work on CorpXData: this comprises search data and taxonomy data from CorpX. We now describe each component of CorpXData.

1. *Search data.* The query-category bipartite graph (Sec. 2.2) as depicted in Fig. 2 is constructed from the query-click logs over $\mathcal{C}$. We have collected query-click data for a set of 55,000 popular queries that had the highest query volume over a period of six months from Jul, 2015 to Dec, 2015. We plan to make this dataset available on the web for academic use.
2. *Taxonomy data.* The taxonomy portion of CorpXData is publicly available. The number of leaf nodes in our taxonomy, $|\mathcal{C}|$, is 986. The number of leaf nodes in the implicit partition of is 149. Namely, each leaf node in the customer-facing taxonomy is a child of one of the 149 parent nodes.

Protocol. We study the islands of interest generated by IslandComms and IslandClusts. In both cases, we vary the number of islands, K, in the range 90–280. We use the implicit partition (recall, this is the partition the taxonomy itself induces) as a baseline. We try to characterize the islands produced by our two algorithms in terms of the four measures—locality, flow, girth, expansion coefficient—introduced in Sec. 5.

Parameter Choices.

1. Value of K is experiment specific. For IslandComms, K is specified implicitly by specifying θ, whereas for IslandClusts, K is specified explicitly.
2. The authors in [20] recommend t to be in the range 3–8, and we observed our results to be stable with respect to the choice of t in the range 4–16. Accordingly, we use $t=4$ for all our experiments, without loss of generality.

7 Empirical Findings

Islands of Interest mined by IslandComms			
	Leaf Categories	ρ	η
1.	hand tools, window repair, masonry contractors, air conditioning, cabinet refinishing, bathroom design, kitchen design, cookware, freezers, ceiling fan installation, light fixture, interior painting, house painting, patio furniture, toilet installation, laundry, carpet cleaning, mattress cleaning, mattress protectors, gutter cleaning, furnace repair, power tools, water heater installation, electrician, roofing contractor, closet	3.02	2.33
2.	natural medicine, tea and lemonade, weight loss, nutritionist, acupuncture, massage, reflexology, shiatsu, facial, day spa, sauna, dental, family doctor, lasik, cosmetic procedures, hair restoration, facial peel, podiatrist, nail salons, mani-pedi, hair perm, tattoo, make-up, dermatologist	2.47	2.60
3.	cocktail mixers, wine coolers, wine bars, wineries, festivals, coffee and treats, ground coffee, tea, coffee capsules, cafes, karaoke bars, whale watching, concerts, pop music, pizza, sushi, thai restaurants, seafood restaurants, mexican restaurants, french restaurants, italian restaurants, cuban restaurants, hot chocolate, pubs, theater and shows, sailing, tours, travel magazines, comedy clubs, amusement parks, rental services, museums, ice skating, acting classes [2]	2.97	2.84
4.	rowing, pilates, health clubs, yoga, fitness classes, gymnastics, dance classes, crossfit, personal trainer, boxing, spinning, zumba, gyms, martial arts	2.23	4.67
5.	cell phone accessories, chargers and adaptors, cell phones, phone cases	1.00	1.75

Islands of interest mined by IslandClusts			
	Leaf Categories	ρ	η
1.	carpet cleaning, hardwood floor cleaning, tile cleaning, carpet installation, carpet repair, hardwood floor repair, hardwood floor cleaning, baseboards	1.70	2.63
2.	carpenters, general contractors, handyman, electricians, drywall repair, ceiling fan installation, light fixture, outlet installation, toilet installation, bathroom remodeling, kitchen remodeling	1.80	3.00
3.	couples massage, deep tissue massage, hot stone massage, pre natal massage, thai massage, reflexology, chiropractor, natural medicine, spa, bath house	1.99	2.50
4.	tattoo removal, hair removal, hair perm, med spa, dermatologist, plastic surgery 5.	2.35	6.36
5.	liquor and spirits, cocktail mixers, dive bars, pubs, wine bars, brazilian restaurants, cuban restaurants, french restaurants, italian restaurants, pizza, seafood restaurants	2.19	2.44
6.	health clubs, pilates, personal trainer, spinning, yoga, zumba, boot camps, gyms, fitness classes	1.96	7.64
7.	cell phones, phone cases, phone batteries, phone accessories, cable chargers and adaptors, tools and equipment	1.52	3.43

Table 1. Examples of high click volume islands of interest mined using IslandComms ($\theta = 0.1$) and IslandClusts ($K = 156$) along with their respective girth (ρ) and expansion coefficient (η). Note the similarity between island 1 from IslandComms with islands 1 and 2 from IslandClusts; likewise, islands 3 and 4 from IslandClusts with island 2 from IslandComms.

7.1 Specificity and similarity in islands of interest

Table. 1 shows a sample of islands of interest having the highest click volume that are mined by IslandComms and IslandClusts. We have chosen $K = 156$ (directly for IslandClusts and indirectly by choosing θ for IslandComms). This is done in order to make the number of islands mined algorithmically close to the number of islands present in the implicit partition induced by the taxonomy itself. We outline our observations below.

1. It can be observed that islands from IslandComms tend to have large variation in sizes, whereas those from IslandClusts tend to have more even sizes.
2. Islands mined by IslandComms and seem to club together categories which are somewhat distinct but may have high similarity in the user communities having very similar nodes.
3. (related to the above) Islands from IslandClusts may bifurcate islands discovered by IslandComms. As an example: Island 2 from IslandComms is a general "beauty" related island. In IslandClusts, we find that there are, in fact, two beauty islands (3, 4)— one centered exclusively on massage and natural therapy, whereas the other is centered on hair treatment and skin cosmetic procedures. Similarly, some of the leaf nodes from the general "home improvement" island (1) from IslandComms seem to have been decomposed into a "service contractor" island (2) and a "floor cleaning/repair" island (1).

7.2 Locality-flow trade-off in islands of interest

Our next objective was to study the locality-flow trade-off for the islands discovered by IslandComms and compare that to those found by IslandClusts. In both cases, we vary the number of islands K. In the case of IslandComms, K is varied implicitly by varying θ in the range [0.02,0.5] (the number of islands varies monotonically with θ in this range— see Fig. 5(a)). For IslandClusts, we specify the number of islands explicitly.

The results are shown in Fig. 5(b). The results clearly indicate that while IslandClusts trades off lower interest flow to achieve higher locality, IslandComms tries to optimize for lower interest flow. This can also be observed in the nature of the communities (Table 1) found by the two algorithms, as discussed in §7.1.

7.3 Girth and expansion coefficient in islands

Next, we show how the islands generated by IslandComms and IslandClusts compare to the implicit partitions of the taxonomy, with respect to the distance measures of girth and expansion coefficient defined earlier. The results are shown in Tables 1 and 2. We find that, in general, the girth of the islands discovered by IslandComms is larger than those discovered by IslandClusts. Namely, there is more diversity in the leaf nodes within an island discovered

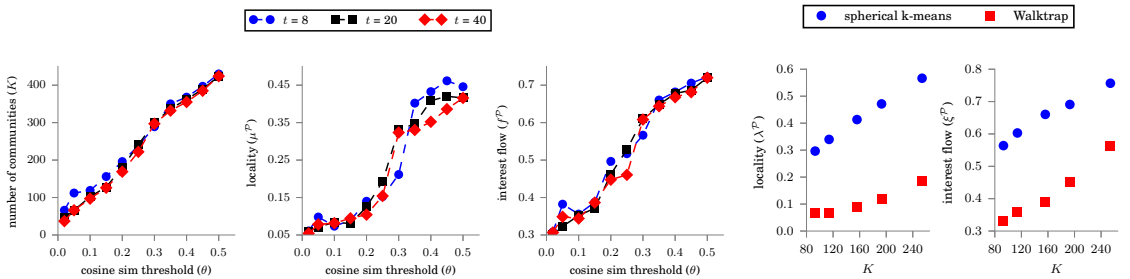

Fig. 5. (a) Variation in number of communities, locality and interest flow as the threshold θ in the cosine similarity projection is varied keeping the number of steps in the IslandComms algorithm t fixed. (b) Variation of locality and interest flow with the number of islands K in IslandComms and IslandClusts.

by IslandComms. However, the expansion coefficient of the islands from IslandClusts is, in general, higher. This shows that IslandClusts selects fewer sister nodes from the implicit partition. This further underlines the "specificity vs. discovery" trade-off between the two algorithms discussed earlier.

	implicit partition	**IslandComms** ($\theta = 0.1$)	**IslandClusts** ($k = 156$)
size (K)	149	156	156
locality (λ)	0.13	0.09	0.41
flow (ξ)	0.88	0.39	0.66
median girth (ρ)	1.00	2.33	2.33

Table 2. Comparison of algorithmically generated partitions (using IslandComms and IslandClusts) with the implicit partition. The size of the partitions is kept similar to facilitate fair comparison.

8 Applications

8.1 Taxonomy Design: How good is the implicit partition?

We compare the implicit partition with the partitioning dictated by the islands mined through IslandComms and IslandClusts. For a fair comparison, we choose parameter settings such that the number of islands K is close to $K = 149$ from the implicit partition. Results are shown in Table. 2. The implicit partition is *sub-optimal* with respect to both locality and flow, as the taxonomy partition generated by IslandClusts not only has a lower interest flow, but more than three times the locality. The partition from IslandComms sacrifices locality to achieve a much lower flow which is a 56% reduction from the flow in the implicit partition. Finally, all the algorithmically generated partitions are quite "far" (in terms of taxonomy distance) from the implicit partition, as indicated by their median girth. Note that the depth of the taxonomy, Δ is four and the high value of their median girth shows that, on average for any two leaf nodes in the algorithmically generated partitions, we would need to move up more than two levels to find a common ancestor.

8.2 Facility Allocation for Inventory: Can islands of interest inform inventory placement?

Items belonging to categories within an island correspond to same interest and therefore have a higher likelihood of being co-purchased. For instance, items in categories "cell phones", "cell phone accessories", and "phone cases" (Island (5) from IslandComms) are likely to be co-purchased together. Hence, placing their merchandise together will result in reduced processing and shipping time and costs. More generally, inventory determined by a given island of interest should be co-located.

8.3 Recommendations for Browse: Can Islands of Interest inform item recommendations?

If a user has expressed interest in a category either through explicitly rating, or through implicitly browsing or purchasing items in that category, then we can recommend items from remaining categories in that island. For instance, if a user shows interest in "window repair" category then she is likely to be interested in items from "hand tools"(Island (1) from IslandComms). In E-commerce, widgets such as "related items", "you may also like", "suggestions based on your last purchase" are widely used to provide recommendations in this manner. This is first of the two ways these Islands of interest thus computed can drive recommendations.

Another way these islands can be used to provide recommendations is through extending collaborative filtering recommender systems. User interests are projected over Island space using implicit user signals such as click or purchase activities, thus generating user profile vector in user-intent space. Then for a given user, other users with like user-intent can be identified using cosine similarity metric, and standard collaborative filtering techniques can be used thereafter to drive recommendations.

Furthermore, we can tune the "discovery vs. specificity" of recommendations by appropriate choice of algorithm and tuning the number of islands in each algorithm. As noted earlier, IslandComms tends to facilitate discovery since islands identified have more diversity as compared to IslandClusts. Conversely, IslandClusts tends to facilitate specificity. In addition, number of islands can also be tuned in each algorithm: higher number of islands tend to make each island more specific in user-intent, thus trading off discovery for specificity.

9 Discussion and Conclusions

In this paper, we construct *islands of interest*—these are sets of leaf nodes of the product taxonomy in e-commerce within which user intent tends to concentrate. In order to mine islands of interest, we have used an assortment of data mining techniques to tap arguably the fastest growing component of e-commerce data, namely search data. We provide two algorithms—IslandComms and IslandClusts—to mine islands of interest. We motivate this new data mining construct using three core e-commerce applications: recommendations, inventory location, and taxonomy design.

In order to analyze and characterize islands of interest in a way that is insightful to these applications, we construct a framework of four measures: locality, flow, girth, and expansion coefficient. Then, we vary parameter values over ranges, and analyze the properties of the resulting islands from IslandComms and IslandClusts using the above four measures. Our conclusions can be summarized as follows.

1. IslandComms and IslandClusts trade off locality and flow very differently. For the same number of islands, IslandClusts optimizes for high locality, whereas IslandComms optimizes for low interest flow.
2. IslandComms tends to club more seemingly diverse leaf nodes into an island, while IslandClusts tends towards more specificity.

We then tie these properties of islands of interest back to our motivating applications. Here we find the following.

1. The implicit partition of the manually generated taxonomy tree is not optimal with respect to locality and flow. That is, there are algorithmically generated partitions—namely, islands of interest—that achieve higher locality and lower interest flow when compared to the implicit partition. Furthermore, these algorithmically generated partitions have high girth which shows that they are not minor rearrangements of the product taxonomy. Islands of interest may be used to replace taxonomic closeness as the building block of recommendations.
2. We can tune the "discovery vs. specificity" by appropriate choice of algorithm (and tuning the number of islands in each algorithm).
3. Islands of interest offer a principled solution to the problem of co-location of inventory.

In future work, we plan to use islands of interest for user modeling: namely, to model users as mixture models over islands of interest.

References

1. D. Beeferman and A. Berger. Agglomerative clustering of a search engine query log. In *In Proceedings of the sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 407–416. Acm Press, 2000.

2. J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez. Recommender systems survey. *Knowledge-Based Systems*, 46:109–132, 2013.
3. P. Boldi, F. Bonchi, C. Castillo, D. Donato, and S. Vigna. Query suggestions using query-flow graphs. In *Proceedings of the 2009 Workshop on Web Search Click Data*, WSCD '09, pages 56–63, New York, NY, USA, 2009. ACM.
4. Y. H. Cho and J. K. Kim. Application of web usage mining and product taxonomy to collaborative recommendations in e-commerce. *Expert Systems with Applications*, 26(2):233 – 246, 2004.
5. Y. H. Cho, J. K. Kim, and S. H. Kim. A personalized recommender system based on web usage mining and decision tree induction. *Expert Systems with Applications*, 23(3):329 – 342, 2002.
6. N. Craswell and M. Szummer. Random walks on the click graph. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, pages 239–246, New York, NY, USA, 2007. ACM.
7. I. S. Dhillon and D. S. Modha. Concept decompositions for large sparse text data using clustering. *Mach. Learn.*, 42(1-2):143–175, Jan. 2001.
8. M. Hepp, J. Leukel, and V. Schmitz. A quantitative analysis of product categorization standards: content, coverage, and maintenance of ecl@ss, unspsc, eotd, and the rosettanet technical dictionary. *Knowledge and Information Systems*, 13(1):77–114, 2007.
9. J. Kamahara, T. Asakawa, S. Shimojo, and H. Miyahara. A community-based recommendation system to reveal unexpected interests. In *11th international multimedia modelling conference*, pages 433–438. IEEE, 2005.
10. Y. S. Kim. Recommender system based on product taxonomy in e-commerce sites. *Journal of Information Science and Engineering*, 29(1):63–78, 2013.
11. T. Lee, I. hoon Lee, S. Lee, S. goo Lee, D. Kim, J. Chun, H. Lee, and J. Shim. Building an operational product ontology system. *Electronic Commerce Research and Applications*, 5(1):16 – 28, 2006. International Workshop on Data Engineering Issues in E-Commerce (DEEC 2005).
12. P. Lops, M. De Gemmis, and G. Semeraro. Content-based recommender systems: State of the art and trends. In *Recommender systems handbook*, pages 73–105. Springer, 2011.
13. O. Matos-Junior, N. Ziviani, F. Botelho, M. Cristo, A. Lacerda, and A. S. da Silva. Using taxonomies for product recommendation. *Journal of Information and Data Management*, 3(2):85, 2012.
14. Q. Mei, D. Zhou, and K. Church. Query suggestion using hitting time. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, CIKM '08, pages 469–478, New York, NY, USA, 2008. ACM.
15. D. Melamed. Community structures in bipartite networks: A dual-projection approach. *PLoS ONE*, 9(5):e97823, 05 2014.
16. A. Nandi and P. A. Bernstein. Hamster: Using search clicklogs for schema and taxonomy matching. *Proc. VLDB Endow.*, 2(1):181–192, Aug. 2009.
17. M. E. J. Newman. Scientific collaboration networks. i. network construction and fundamental results. *Phys. Rev. E*, 64:016131, Jun 2001.
18. M. OConnor and J. Herlocker. Clustering items for collaborative filtering. In *Proceedings of the ACM SIGIR workshop on recommender systems*, volume 128. UC Berkeley, 1999.
19. S. Park and W. Kim. Ontology mapping between heterogeneous product taxonomies in an electronic commerce environment. *International Journal of Electronic Commerce*, 12(2):69–87, 2007.
20. P. Pons and M. Latapy. Computing communities in large networks using random walks. *J. of Graph Alg. and App. bf*, 10:284–293, 2004.
21. S. Sahebi and W. W. Cohen. Community-based recommendations: a solution to the cold start problem. In *Workshop on recommender systems and the social web, RSWEB*, 2011.
22. B. M. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Recommender systems for large-scale e-commerce: Scalable neighborhood formation using clustering. In *Proceedings of the fifth international conference on computer and information technology*, volume 1, 2002.
23. X. Su and T. M. Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in artificial intelligence*, 2009:4, 2009.
24. B. Xu, J. Bu, C. Chen, and D. Cai. An exploration of improving collaborative recommender systems via user-item subgroups. In *Proceedings of the 21st international conference on World Wide Web*, pages 21–30. ACM, 2012.
25. T. Zhou, J. Ren, M. c. v. Medo, and Y.-C. Zhang. Bipartite network projection and personal recommendation. *Phys. Rev. E*, 76:046115, Oct 2007.
26. C.-N. Ziegler, G. Lausen, and L. Schmidt-Thieme. Taxonomy-driven computation of product recommendations. In *Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management*, CIKM '04, pages 406–415, New York, NY, USA, 2004. ACM.
27. K. Zweig and M. Kaufmann. A systematic approach to the one-mode projection of bipartite graphs. *Social Network Analysis and Mining*, 1(3):187–218, 2011.