

Interactive Critiquing for Catalog Navigation in E-Commerce

Robin Burke

Department of Information Systems and Decision Sciences

California State University, Fullerton

phone: 714-278-5513 / fax: 714-278-5940

rburke@fullerton.edu

Keywords

case-based reasoning, electronic commerce, information retrieval, recommender systems

Abstract

E-commerce sites can have large, essentially unbounded, catalogs. With large catalogs comes increasing difficulty for buyers in making use of standard search and browsing facilities. Particularly in the case of casual or occasional buyers and in the case of complex products, the gap between a product's specifications and the buyer's understanding of need can be hard to bridge. An effective e-commerce catalog must map user needs to products that can fulfill them. This paper describes an interactive, incremental, case-based, critiquing approach to solving this problem. The approach is interactive and incremental, so it does not require that the user have a completely specified need at the start. The system is case-based in that it emphasizes products over features or constraints, and uses case-based reasoning techniques for its product retrieval. Finally, the approach is based on the critiquing of presented examples, each critique redirecting the search to home in on appropriate products.

1. Introduction

Perhaps the most public manifestations of electronic commerce are the big business-to-consumer sites that draw millions of visitors: Amazon.com, CDNow, and others. One of the key advantages enjoyed by these retailers is inventory. Unlike a brick and mortar store, which is limited by shelf space, and paper catalogs, which are costly to produce, ship and keep updated, an electronic catalog can be essentially unlimited in size and is always current. However, this endless bounty of goods raises a problem for consumers: how to find desired products? Users of large catalogs may feel as if they are wandering an immense space with little guidance.

Keyword and parametric search and category-based browsing are the dominant solutions for finding products, but both users and retailers find them unsatisfying (Hagen, et al. 2000). Indeed, large on-line retailers are constantly adding new technologies to augment search. Collaborative filtering and other recommendation mechanisms are commonly employed (Schafer, et al. 1999). In order to improve search itself, we need systems that understand users' needs and mediate between those needs and the

representation of products. Some sites are beginning to use similarity-based retrieval techniques from case-based reasoning (CBR), which are equipped to perform more flexible needs-based retrieval.

Search by itself is not enough, however. When search is the only navigation mechanism, users are forced to repeat the search operation over and over with no guarantee of getting closer to their goal (Hagen et al. 2000). This paper demonstrates a critique-based navigation method that preserves the knowledge-based benefits of CBR retrieval but adds navigation, providing a natural next place to go when search fails.

We have described the underlying technology of critique-based navigation in other publications (Burke et al. 1997; Burke, 1999; Burke, 2000). The purpose of this paper is to explore the consequences of this technique for e-commerce catalogs and to compare this approach with approaches based on requirements gathering. We argue that interactive critiquing is more natural, requires less user effort and as a side benefit generates semantically-rich profile data is useful for personalizing product presentation and to analyzing catalog contents.

2. Existing catalog search technologies

Catalog search comes in several varieties. Keyword search draws from information retrieval and is text-based. The familiar search box found on e-commerce sites connects the user with this type of search. Where technical criteria are important, a catalog may use instead parametric search where the user supplies specific parameters that products must meet.

2.1. Keyword search

Keyword-based catalog search is an adaptation of information retrieval technology to the problem of product location. Products are typically ranked so that those containing more of the keywords in the user's query come first. The application of information retrieval to catalog search has not been entirely successful. It is well known that document retrieval systems are sensitive to the amount of text in each document. The larger the document, the more reliable retrieval can be. Catalog descriptions are invariably short, less than several hundred words (sometimes much less) and so are simply too impoverished for effective retrieval.

The inadequacy of the text associated with products intensifies the well-known problem of terminological mismatch in information retrieval (Foskett, 1980). If documents are large, they are likely to contain a greater diversity of terms, but if they are short, the chances are increased that the terms a user uses will not match those with which the product is described.

If the user's language does not match the vocabulary of the catalog, little of use can be found. For example, a recent search for "new cpu for pc" in Amazon.com's book catalog turned up no computer-related items, while a search using the correct terms "upgrade pc" resulted in 31 highly-relevant books being found. If we can think of the product being sought as a solution and the customer's need as a problem to be solved, keyword search requires that the customer be able to create or envision the solution in order to describe its parameters to the search system. Some needs can be easily "reverse-

engineered”. A user who wants the latest book by Stephen King might try “Stephen King 2001 2002”, for example. Other needs require expertise: a user who only knows that he wants a relaxing island vacation will not get much help from a travel search site.

2.2. Parametric search

Search can be constrained to more directly match the user’s goal. For example, music sites offer more sophisticated search in the classical music arena, so that a user can search for performances of Brahms’ “Variations on a Theme by Haydn” without getting Haydn’s own works in the mix. In more technical products, this kind of feature-oriented search organizes the user’s query along the lines of the features with which products are described. Ultimately, one ends up with parametric search, in which the query consists of selected features or feature ranges for database querying.

If we view the set of available products as a large multi-dimensional space, searching (especially parametric search) is essentially a narrowing operation that extracts a subset from the original space and displays it, giving a closer-up view. If the original space is very large, the new space may contain too many products to comfortably scan, possibly calling for a repeat of the same search/narrowing operation. Iterative refinement is possible in hierarchical catalogs such as eBay and Yahoo, but in most product catalogs it is not, and one is left either to wade through 1000 books on “electronic commerce” or to reformulate one’s query and try to arrive at a more manageable result. This process of narrowing and backing out is a major part of the frustration that users express regarding product search (Hagen, et al. 2000).

Even if iterative refinement is possible, a catalog that relies entirely on search has limited effectiveness because it does not support the complete product selection process. A search-focused approach begins with the assumption that buyer has a fixed goal or requirement in mind, and that she will formulate searches based on this fixed requirement until the need is satisfied. However, it is also the case that searchers learn from searching and they may adjust their goals as the process continues. A user might find, for example, that there are no apartments under \$1000 for rent in a certain area, and have to decide either to pay more or to try other locations. A system that operates by narrowing makes such “dead-ends” particularly frustrating, since a null result gives no information about nearby products.

The narrowing process makes it difficult for users to perform comparisons between items that the system has placed in different categories. For example, the Realtor.com site organizes real estate listings by zip code. Two houses in adjacent zip codes might be across the street from each other, but they will not appear in the same search unless all of both zip codes are also included, doubling the size of the search space. The problem is that the user’s category (geographic proximity) does not match the database’s approximation of it: the zip code. One might argue that such mismatches are inevitable, a natural consequence of limited data and the wide range of users’ interests and needs. Our experience, however, indicates that this is not so. If the catalog presents a user-oriented interface rather than a data-oriented one, it can capture user needs more directly, and an intelligent system can mediate between the user’s expression of need and the products available to satisfy it.

2.3. CBR retrieval

One natural analogy for the product selection process is the problem of case retrieval in a case-based reasoning system. A case-based reasoning (CBR) system is a problem solver that uses the recall of examples as the fundamental problem-solving process (Riesbeck and Schank, 1989; Kolodner, 1993). When confronted with a new problem, the CBR problem solver examines its memory for past solved problems that are similar. It then applies the old solution to the new problem, adapting as necessary. Case retrieval therefore requires that the system match a new problem against a known solution. If we replace “new problem” with “user need” and “known solution” with “product,” this is precisely what an e-commerce catalog should do.

Case retrieval is knowledge-based, so it is more adaptable to vocabulary mismatch than keyword techniques. It is also ranked retrieval, with results ordered by “goodness of match” to the goal (Watson, 1997). This makes case retrieval approaches less vulnerable to the null result problem. The user’s request is treated as a starting point rather than a set of absolute constraints and the constraints can be relaxed to find, in the case of the real estate example, slightly more expensive apartments or ones in a nearby neighborhood. The case-based retriever looks for the closest cases to some central query point. Closeness is not defined strictly in terms of the feature space, but is subject to an arbitrary similarity function.

The case retrieval approach to e-commerce catalog search has demonstrated its effectiveness in research prototypes and commercial deployments, and several organizations market case retrieval products tailored specifically for e-commerce.¹ These systems go a considerable distance toward solving the problem laid out above. The user can present a problem or need as input and the system will match it against products, without the user having to map his or her need into an arcane feature vocabulary. The effectiveness of such a solution depends, as always in knowledge-based systems, on the quality of the knowledge engineering effort that supports it.

2.4. The need for navigation

However, retrieval – even knowledge-based retrieval – is still subject to the narrowing problem described above. If the result set does not contain a product of interest, the user has little choice but to begin again with a reformulated search. What is needed is a means of navigating through the contents of the catalog, starting from a location in the product space determined by a search. With such a function, we can avoid the narrowing trap, and provide a reasonable next place for the user to go, if the current item is not satisfactory.

The question then becomes how should products be related in a catalog for navigational purposes? In the domain of intelligent tutorial systems, researchers in “ASK systems” (Ferguson, et al. 1992) have found that tutorial cases can be related by certain standard question types: case A answers a question that case B raises. We have found that an effective way for products to be related is by *critique*, by goals that are not satisfied by

¹ See empolis’s orange (<http://www.empolis.com/englisch/home.htm>), and Kaidara’s Kaidara Commerce product (<http://www.kaidara.com/>).

the current item. A system organized in this way allows the user to express her dissatisfaction with a product precisely at the moment when that dissatisfaction is felt. Figure 1 shows how this navigation is achieved. The user searches and finds a set of products in a particular region of the product space as represented by the oval.

When the user sees that the results of the original search are inadequate, she can redirect the search by a navigation action based on the inadequacies identified. If the problem is that the examples show too much of Feature A, a “Too Much A” critique can be used to push the search to a different part of the space. This design ameliorates the narrowing problem and unifies search and navigation, with search providing a point of entry exploited by the navigational operation.

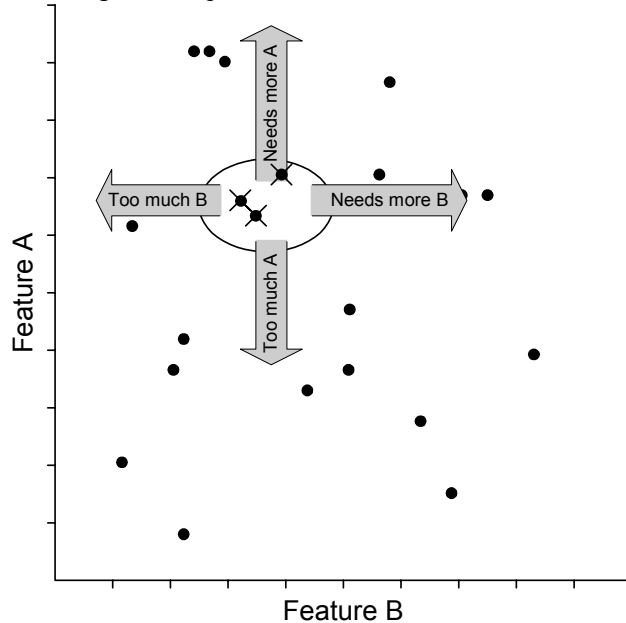


Figure 1. Critique-based navigation as a solution to the narrowing problem.

For some products, it is possible even to avoid search altogether. Some domains such as restaurants and movies contain products that are uniquely identified by name. Named products provide the opportunity to jump into the navigation space without having to input a set of desired features and indeed, without the user even having to articulate her need in terms of features. A viewer might wish to rent a movie just like a film he has just seen in the theater, for example.

3. The Entree restaurant recommender

These two catalog access methods, the named-product entry point and critique-based navigation, are implemented in the Entree restaurant recommendation system, a system that emerged from a line of research known as “FindMe Systems” (Burke et al. 1997; Burke 1999; Burke, 2000). Entree was implemented to serve as a restaurant guide for

attendees of the 1996 Democratic National Convention in Chicago and remains in public operation.² A user interacts with the system by submitting an entry point, either a known restaurant or a set of criteria, and is presented with similar restaurants. The user interacts with the system in a dialog, critiquing the system's suggestions and interactively redirecting the search until an acceptable option is achieved.

Consider a user who starts the interaction by entering a query in the form of a known restaurant, Wolfgang Puck's "Chinois on Main" in Los Angeles. As shown in Figure 2, the system finds a similar Chicago restaurant that combines Asian and French influences, "Yoshi's Cafe," as well as other similar restaurants that are ranked by their similarity. Note that the connection between "Pacific New Wave" cuisine and its Asian and French culinary components is part of the system's knowledge base of cuisines. The user, however, is interested in a cheaper meal and selects the "Less \$\$" button. The result shown in Figure 3 is a creative Asian restaurant in a cheaper price bracket: "Lulu's." However, the French influence is lost — one consequence of the move to a lower price bracket. The user can continue critiquing until an acceptable restaurant has been located.

The screenshot shows a web interface titled "Entree Results". It displays the details of a selected Los Angeles restaurant, "Chinois On Main", and a recommended Chicago restaurant, "Yoshi's Cafe". Below the recommendations are several circular buttons for filtering: "less \$\$", "nicer", "cuisine", "traditional", "creative", "livelier", and "quieter". At the bottom, there is a section titled "For other suggestions, select:" followed by a grid of restaurant names with red underlines, including Yoshi's Cafe, Penny's Noodle Shop, Emilio's Tapas Bar & Restaurant, Emilio's Granada, 302 West, Arun's, Nick's Fishmarket, Lulu's, Trio, and Bossa Nova.

Entree Results

The Los Angeles restaurant you chose is:

Chinois On Main	
2709 Main St. (bet. Rose Ave. & Ocean Park Blvd.), Santa Monica, 310-392-9025	
Pacific New Wave	\$30-\$50
Extraordinary Decor, Extraordinary Service, Near-perfect Food, Hip Place To Be, On the Beach, Great for People Watching, Parties and Occasions, Weekend Brunch, Weekend Lunch, Fabulous Wine Lists	

We recommend:

Yoshi's Cafe	
3257 N. Halsted St. (Belmont Ave.), Chicago, 312-248-6160	
Asian, Japanese, French (New)	\$30-\$50
Extraordinary Decor, Extraordinary Service, Near-perfect Food, Need To Dress, Prix Fixe Menus, Quiet for Conversation, Very Busy - Reservations a Must, Romantic, Good Out of Town Business, Fabulous Wine Lists, Game, Parking/Valet	

[less \\$\\$](#) [nicer](#) [cuisine](#)
[traditional](#) [creative](#) [livelier](#) [quieter](#)

For other suggestions, select:

Yoshi's Cafe	302 West	Lulu's
Penny's Noodle Shop	Arun's	Trio
Emilio's Tapas Bar & Restaurant	Nick's Fishmarket	Bossa Nova
Emilio's Granada		

Figure 2. Similarity-based Recommendation

² <http://pippen.infolab.northwestern.edu/entree>

Entree Results

For a cheaper restaurant than:

Yoshi's Cafe	
3257 N. Halsted St. (Belmont Ave.), Chicago, 312-248-6160	
Asian, Japanese, French (New)	\$30-\$50

We recommend:

Lulu's (map)	
626 Davis St. (bet. Chicago & Orrington Aves.), Evanston, 708-869-4343	
Japanese, Asian	below \$15
Good Decor, Excellent Service, Excellent Food, Creative, No Reservations, Weekend Brunch, Wheelchair Access, Long Drive	

less \$\$ *nicer* *cuisine*

traditional *creative* *livelier* *quieter*

For other suggestions, select:

<p>Lulu's</p> <p>Noodle Noodle</p> <p>New Japan</p> <p>Kampai</p>	<p>Penny's Noodle Shop</p> <p>Benihana of Tokyo</p> <p>Hatsuhana</p> <p>Aki Hana</p>	<p>Sanko</p> <p>Honda</p> <p>Daruma</p>
---	--	---

Figure 3. Critique-based Navigation (“Less \$\$”)

A key consideration in the design of Entree was to support a natural interactive retrieval process. After the first page, no action requires more than a single click, an important consideration in efficient use of the web medium. The system presents one main result and a small number of neighbors rather than an overwhelming list, and the user can explore the space of restaurants, discovering for example, the tradeoffs between price and quality for restaurants serving a given cuisine.

3.1. FindMe retrieval

The FindMe technique is one of knowledge-based similarity retrieval. There are two fundamental retrieval modes: similarity-finding and critique application. Similarity finding implements named product retrieval. The user selects a given item from the catalog (called the *source*) and requests other items similar to it. A case retrieval is performed, looking for those items most similar to the source. Figure 4 shows this retrieval schematically. It differs only from Figure 1 in that an actual product is used as the starting point of the retrieval.

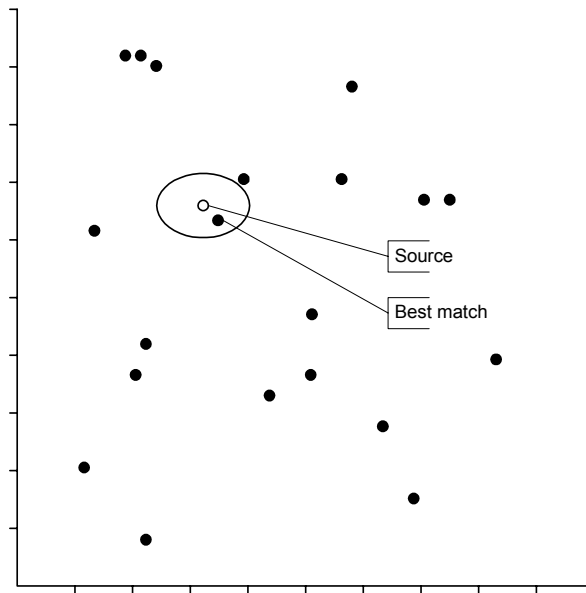


Figure 4. Named-product retrieval in FindMe systems.

Critique application is similar except that the candidate set is filtered prior to sorting to leave only those candidates that satisfy the critique. For example, if a user responds to the restaurant “Lulu’s” with the critique “Nicer,” the system determines the “niceness” value of “Lulu’s” and rejects all candidates except those whose value is greater. Figure 5 shows this retrieval method in action. The user has selected “More B” so the retriever filters out all items with the same level or lower level of feature B as the source (the gray area).

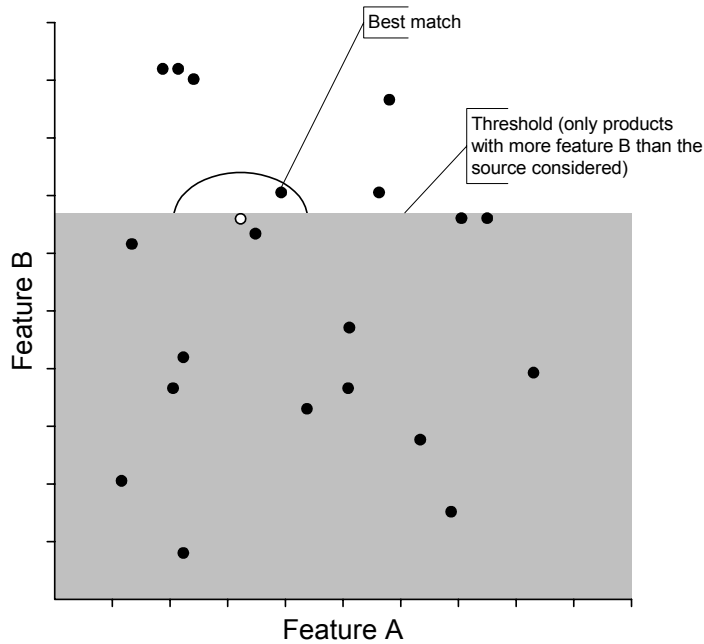


Figure 5. Critique-based navigation in FindMe systems.

Our initial FindMe experiments demonstrated something that case-based reasoning researchers have always known, namely that similarity is not a simple or uniform concept. In part, what counts as similar depends on what one's goals are: a car is similar to a boat if one's destination is near water, but not if it is landlocked. FindMe similarity measures therefore have to be goal-based, and consider multiple goals and their tradeoffs. Typically, there are only a handful of standard goals in any given product domain. For each goal, we define a *local similarity metric*, which measures how closely two products come to meeting the same goal. Two restaurants with the same price range would get the maximum similarity rating on the metric of price, but may differ greatly on another metric, such as quality or type of cuisine.

Through our FindMe prototypes, we looked at the interactions between goals, and experimented with combinations of metrics to achieve intuitive rankings of products. We found there were well-defined priorities attached to the most important goals and that they could be treated independently. For example, in the restaurant domain, cuisine is of paramount importance. Part of the reason is that cuisine is a category that more or less defines the meaning of other features – a high-quality French restaurant is not really comparable to a high-quality burger joint, partly because of what it means to serve French cuisine.

We can think of the primary category as the most important goal that a recommendation must satisfy, but there are other goals that must be factored into the similarity calculation. For example, in Entree, the goals were cuisine, price, quality, and atmosphere applied in rank order, which seemed to capture our intuition about what was important about restaurants. The hierarchical nature of the global similarity metric makes

the operation of the system easy to engineer, understand and modify. As discussed below, it also makes the output less sensitive to the precise quantitative formulation of the local metrics. The commercial successors of FindMe contain global similarity metrics that are partial orderings of local metrics. In this formulation, local metrics of equal rank are combined and given equal weight, allowing for more flexibility in implementation. This was found to be helpful in the lower priority parts of the metric where comparisons are based on attributes with only a weak influence on similarity and that are difficult to prioritize in any principled way.

It is of course possible that different users might have different goal orderings or different goals altogether. A FindMe system may therefore have several different *global similarity metrics*, combining local measures in different priorities. We can think of the different global metrics as strategies for retrieval, each capturing a different notion of overall similarity.

3.2. Entry points

In order to critique, the user has to be somewhere, looking at some product. We have discussed what we call the *named entity* approach where the user specifies a known product as a starting point for finding something new. This only works when products have this kind of name recognition, as in the case of restaurants, books, and movies. Apartments, on the other hand, do not have this property. The second kind of entry point is case-based *search* of the type introduced in section 2.3. A final entry point possibility is the *stereotype* method in which some set of generic products are chosen that represent the range of possibilities: for example, if a user is buying a car, the stereotypes presented might be a minivan, a four-door sedan, a two-door economy car, an SUV, and a sports car. Ideally, the entry point stereotypes should be presented generically rather than as a specific make and model, using cartoons for example. The user picks the generic product that is closest to what he wants. Just as with the other methods, the purpose of the stereotype is not necessarily to find exactly the right item in the initial retrieval, but to put the user into the product space so that navigation is enabled.

While formal user evaluations on FindMe systems have yet to be performed, we have found broad user acceptance of the systems, both as research prototypes and in commercial spin-offs. In our implementations, we have identified two important success factors:

Product data: Probably the most important success factor is the underlying product data that the system has to work with. The catalog underlying the system must be large enough so that search is a necessity. It must contain sufficient detail to support the inferences that the system needs to make. Also, the catalog must be sufficiently homogeneous that items can be compared against each other.

Knowledge engineering: Knowledge engineering is also important, and in the case of FindMe systems, it is the end-user customers who are the “experts.” Users are the ones responding to the possible critiques that the systems makes available: “nicer,” “cheaper,” etc. It is crucial that these choices be natural ones. In addition, users will be turned off if the results returned do not capture their intuitive sense of what ought to be considered similar, so the similarity metrics must reflect buyers’ understanding of the product space.

The interaction between the product data and knowledge engineering is also important. In some product domains, there may be a straightforward match between product data and the inferences needed to support recommendation. For example, the system will need to make inferences about cost but it can rely on the fact that every product will have an associated price. In other cases, more knowledge is needed in order to bridge the gap. For example, a user may wish to critique movies on complex criteria: asking for a movie “like ‘Basic Instinct’ but funnier”. In this case, the system needs to be able to use whatever information is available about a movie to determine how funny it is likely to be. Keywords and their meanings, features extracted from plot descriptions, and knowledge of actors’ typical roles are all clues that have been used in FindMe movie recommendation prototypes.

3.3. Support for the buying process

FindMe systems do not accumulate a profile that represents the system’s view of the user’s requirements. We view the task as navigating to a product, not building an internal representation. Early FindMe systems did experiment with building short-term models of the user’s goal by building constraint structures based on critiques. This was found to negate the exploratory experience afforded by the system. Users wanted to explore the product space, but found themselves trapped, “narrowed,” by their earlier decisions. An accumulated profile assumes a static set of requirements, not one that may be constantly shifting as the user learns.

We do not assume a “waterfall” model of purchasing: assess need, determine requirements, and then examine products. This model is inadequate to explain much buying behavior. Indeed, current models of buying behavior involve feedback between the search process and the recognition of the problem for which a product is sought (Mullen & Johnson, 1990). By explicitly allowing for this kind of feedback, the FindMe architecture supports more of the buying process than can a system that builds a set of static requirements.

By supporting the feedback between experience and need formulation, FindMe systems help a user incrementally discover his requirements. Each example that the catalog shows forces the user into an evaluation. Unlike the like/dislike binary evaluation offered by relevance feedback (Salton and McGill, 1983), the critiques offered in a FindMe system focus the user’s attention on relevant attributes of the product and make the evaluation of the importance of these attributes the user’s central concern. As a result, the user might find that having a fancy meal tonight is not as important as having a creative one, a fact that he might not have realized without prompting.

3.4. Issues

Several problems remain in the deployment of the FindMe technique in electronic commerce catalogs. Knowledge engineering is obviously a crucial challenge. The local and global metrics must reflect users’ intuition about what counts as similar. The critiques must reflect the dimensions of user’s high-level concerns regarding the various products. Some domains are much easier to engineer than others. With restaurants, a handful of critiques seemed to cover users’ concerns quite well. However, in a more

complex domain, such as movies, one might anticipate wanting to handle a whole universe of possible critiques from the genre-level (“Fatal Attraction” but funnier) to the extremely specific (“Die Hard” without Bruce Willis). Since all possible dimensions of critique must be anticipated by the knowledge engineer, it becomes a pragmatic decision to select and implement those of the greatest value to the greatest number of users.

Similarly, the local similarity metrics must accurately represent users’ sense of similarity. A color-blind user, for example, might have different notions of similarity between color choices than one with normal eyesight. The knowledge engineer again has to attempt to satisfy the greatest number of potential users with local metrics that capture the most common preferences. This problem is ameliorated somewhat by the role of local metrics. Utility-based product recommenders (Guttman, 1998; Maes, et al. 1999) and most case-based recommenders combine local metrics quantitatively and therefore must be concerned about the precise shape and scale of each. By contrast, FindMe metrics are combined hierarchically and what is important is the partial order that each creates. In fact, one can re-envision the retrieval process as a one of ordering the case base rather than measuring distances within it (Ferguson & Bridge, 2000). What is important then, is not the precise details of a local similarity metric, but its general shape and its place within the global metric.

The filtering mechanism for critiques depicted schematically in Figure 5 works well when the domain is reasonably sparse. A problem arises when the domain is dense. Products that marginally meet the critique are not really very different from the source, making navigation to a truly different area of the product space quite slow and frustrating. Figure 6 shows this situation. The user selects an item and asks for “More B”. The next product found is only slightly different from the critiqued item, and subsequent critiques produce only slow change in the quantity of feature B.

Ideally, the system should move along the critiqued dimension just far enough to obtain a difference that is significant to the user. Products found beyond this point will be responsive to the critique without being unduly distant from the starting point. Ultimately, the question of where this boundary lies can only have a domain-specific answer. For commercial applications of the FindMe technology, a set of tools for engineering the product feature space were developed, including statistical summarization and visualization tools. Partly these tools helped assure data quality, but they also had the purpose of allowing the manipulation of the attribute space to achieve meaningful navigation.

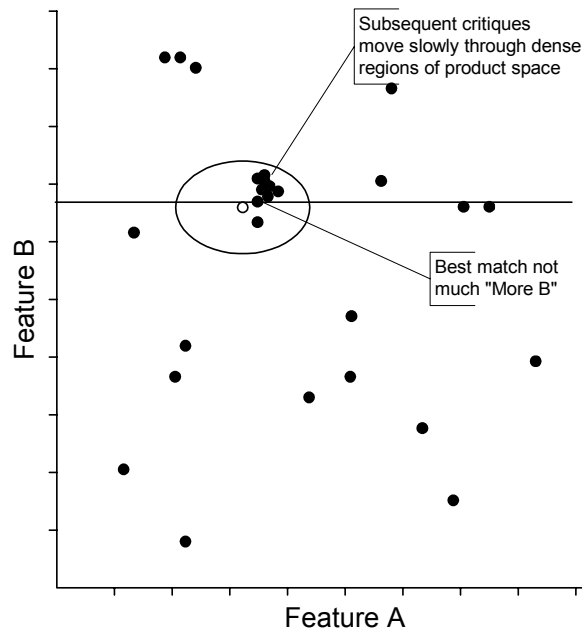


Figure 6. Navigation slowed in dense product space.

The issue of navigation speed highlights the tension between the central notions in FindMe systems: similarity and critique. The user tells the system: “Give me something like this in every way, except different along this attribute.” Of course, not every combination of features is found in the product space, so navigating along one dimension inevitably yields products that are different along other dimensions as well. A critique that moves very slowly in the product space preserves the similarity of the results to the original starting point but does not do justice in responding to the user’s request. On the other hand, a critique that moves radically across the product space abrogates the goal of similarity in the service of finding something different. Neither is desirable. This tradeoff is similar to that described by Smyth and McClave (2001) in their discussion of similarity and diversity as conflicting but desirable properties in case retrieval. Smyth and McClave use a quality measure that incorporates both similarity and diversity measures into their retrieval algorithm to achieve both goals simultaneously. This would be an option for FindMe systems as well, instead of using a filter to satisfy critiques, to employ a quality metric that measures both how well the retrieved item satisfies the critique and how similar it is to the starting point. Another approach would be to add magnitude as well as direction to a critique. That is, make it possible to say “a lot cheaper” as well as just “cheaper.” With this level of control the user could jump more quickly across the product space. We are investigating how such scaled critiques might be incorporated into the FindMe framework.

One weakness that has been noted in the FindMe approach has been its incompatibility with low-level constraints. For example, it is impossible to restrict one’s search just to northern Italian restaurants in Entree, while such a constraint would be easy

to formulate with parametric search. This is a side-effect of the case retrieval technique, which looks for nearby products and does not restrict itself only to the features input by the user. It is also the product of design decisions made when the knowledge engineering for Entree was performed. The “soft” nature of the interaction with a FindMe system may make it less appropriate for technical users who have strict constraints that they are seeking to meet. Such users may also have idiosyncratic preferences over the features in the domain that do not match the general global similarity metrics built into the FindMe system. One possible approach is to let users customize the global similarity metric to suit their tastes. Another would be to let users select attributes in the query to be treated as “hard” constraints.

4. Semantic Ratings

One interesting side-effect of the FindMe navigation technique is the accumulation of critiques associated with different products in the catalog. These critiques are in essence qualitative, semantic ratings, as distinct from the scalar numeric ratings used in collaborative filtering systems. They carry reasons behind preferences, rather simple valence and degree of preference. From these critiques, characteristics of both users and products can be gleaned. For example, the system might be able to identify the gourmards in the database by looking for users who use the “nicer” critique frequently and the “cheaper” critique rarely. The system could identify the over-priced restaurants as being those that get the “cheaper” critique more frequently than might be expected. We have conducted experiments on these types of data mining using user experience with Entree over its first 3 years of public web use, and found that Entree’s semantic ratings could be used for comparing both users and products.

4.1. Collaborative recommendation

Collaborative recommender systems aggregate ratings or recommendations of objects, recognize commonalities between users on the basis of their ratings, and generate new recommendations based on inter-user comparisons. For some of the early systems in this area, see (Goldberg et al. 1992; Resnick et al. 1994; Hill et al. 1995; Shardanand & Maes, 1995; Konstan, et al. 1997).

If we accept critiques as implicit ratings, they can be easily turned into numeric scores of the type used by collaborative algorithms. Many collaborative systems work with simple positive/negative ratings: in our case, critiques would be negative and entry points positive. With numeric values instead of ratings, we can compare two users using standard collaborative filtering algorithms such as Pearson’s correlation coefficient.

Although a simple numeric conversion of the navigation actions into ratings satisfies the input criteria of collaborative filtering algorithms, it is ultimately unsatisfying from a user modeling point of view. The conversion loses entirely the reason for the user’s preference, which has been explicitly given to the system. Suppose Alice looks at “Yoshi’s Cafe” and selects the “Less \$\$” button and Bob looks at the same restaurant and clicks on “Nicer”. The implicit rating approach would count these ratings as evidence that Alice and Bob are similar users, something that the semantics of the ratings would suggest is unlikely.

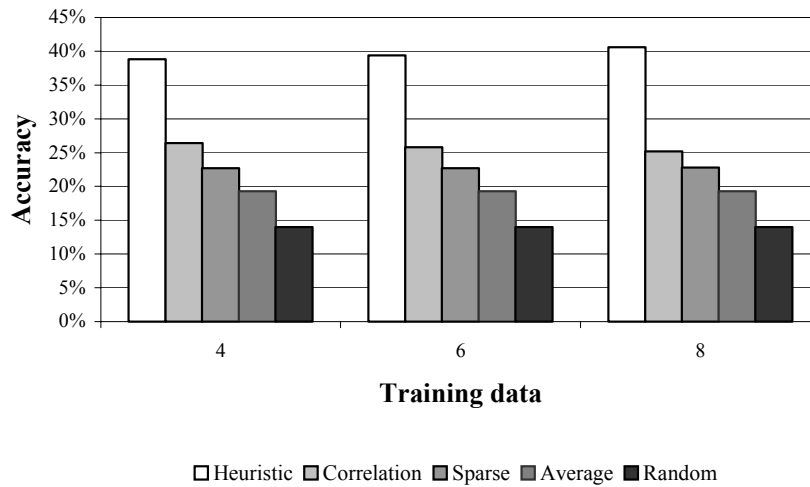
To avoid losing the semantic details of ratings, we can perform collaborative filtering by looking only for users with exactly the same ratings: treating ratings on different dimensions as incommensurable. We would only match Alice against others who also thought that “Yoshi’s Cafe” was too expensive. The problem then becomes a multi-dimensional vector-matching problem of the type frequently encountered in information retrieval contexts (Salton & McGill 1983). We convert the session to a vector where each restaurant / rating combination is a separate dimension. Alice’s session will show a 1 in the position corresponding to the “Yoshi’s Cafe”/”Less \$\$” combination, and Bob’s session would have a 0 there and a 1 in the “Yoshi’s Cafe”/”Nicer” dimension. With each user represented as a multi-dimensional vector, we can use the cosine measure of similarity.

However, this approach yields a high-dimensional vector space and a highly sparse comparison problem. Because we will only compare users whose reactions are largely identical, predictions will be based on a much smaller number of users than in the single-scale collaborative approach.

A final possibility is to take into account the semantics of the ratings themselves: we establish a similarity metric between ratings based on their characteristics. In the example above, we should probably rate Alice and Bob as dissimilar even though they both disliked the same restaurant – they did so for essentially opposite reasons. This is a “heuristic similarity” approach, drawing again on ideas from case-based reasoning. It does not establish a single numeric scale to which all actions are converted, but rather looks at the similarity of users on a rating by rating basis. This metric takes the qualitative differences between ratings into account, but it allows more kinds of inter-user comparison than the sparse metric.

Heuristic comparison preserves the meaning of the critiques more strongly than either of the other collaborative techniques. A series of experiments were conducted exploring the potential for hybrid recommendation combining FindMe knowledge-based recommendation with collaborative recommendation using the different collaborative techniques described above. The experiments used data from logs of user interactions with Entree, using short-term session data only. Each user’s interactive session was partitioned into training and test data, with the evaluation task being to identify a positively-rated restaurant from among other negatively-rated ones in the test set. The training data served as a user’s profile for the purposes of collaborative matching. Additional details can be found in (Burke, in press).

Our experiments demonstrated that, in fact, semantic ratings, when combined using this heuristic similarity approach gave more accurate recommendations than the other collaborative techniques. Figure 7 shows the results from one of the experiments comparing the accuracy of different collaborative algorithms at predicting user preferences in the Entree data. The heuristic technique performs better than either the correlation or vector (sparse) methods. Overall average and random selection are included for reference.



**Figure 7. Comparing Collaborative Algorithms, Session Size = 15
(5x5 cross-validation, n=4,632)**

The other significant recommendation paradigm besides knowledge-based and collaborative recommendation is content-based recommendation, in which the system learns a classifier for each user based on the features of liked and disliked products. We are currently exploring content-based hybrids with FindMe systems and plan to evaluate the benefit of semantic ratings in this context as well.

5. Related Work

The INRECA project sought to develop efficient case retrieval technology for use in a variety of applications including what would now be called e-commerce catalog search (Bergmann, et al. 1999). Two commercial spin-offs from INRECA are currently marketing case retrieval systems for e-commerce catalogs: empolis (formerly tec:inno) and Kaidara (formerly Acknosoft). These systems package the case retrieval technology discussed earlier for finding products based on proximity to a query with knowledge engineering tools for handling catalog data and developing the necessary similarity metrics. More recently, another European project WebSell took some of these same technologies and applied them more specifically to e-commerce applications (Vollrath, et al. 1998).

Another approach to the problem of catalog search comes from the discipline of decision theory. From multi-attribute utility theory comes the notion of making decisions by combining and trading off between different goals expressed in the form of utility functions (Keeney & Raiffa, 1993).

Utility-based catalog search follows the waterfall model, establishing requirements up front, but rather than representing requirements as specific values for features, the user

creates a utility function for each feature. The search process therefore becomes one of ranking products by their utility under the functions that have been defined. One benefit of this approach is that it becomes possible to represent non-product features such as delivery schedule under the same rubric as product features letting a user with a crucial deadline easily factor it into the decision. The Tête-à-tête system was among the first to use this approach (Guttman, 1998; Maes, et al. 1999).

Information retrieval research in general focuses on search, but some researchers have explored navigational capabilities through interactive retrieval clustering (Cutting, et al., 1992) and through relevance feedback (Salton & McGill, 1983). Our approach differs from relevance feedback approaches in both explicitness and flexibility. In relevance feedback, the user selects some retrieved documents as being more relevant than others, but does not give any detailed feedback about the reasons why. In FindMe systems, critiques supply concrete domain-specific feedback. In addition, FindMe systems are not limited to finding items based on similarity alone. The user does not say "Give me more items like this one," the aim of relevance feedback and clustering systems, but instead asks for items that are different from a presented item in some particular way.

Schneiderman's "dynamic query" systems present another approach to database navigation (Schneiderman, 1994). These systems use two-dimensional graphical maps of a data space in which examples are typically represented by points. Queries are created by moving sliders that correspond to features, and the items retrieved by the query are shown as appropriately-colored points in the space. This technique has been very effective for two-dimensional data such as maps, when the relevant retrieval variables are scalar values. Like FindMe, the dynamic query approach has the benefit of letting users discover tradeoffs in the data because users can watch the pattern of the retrieved data change as values are manipulated. However, as we found in our early FindMe experiments, direct manipulation is less effective when there are many features to be manipulated, especially when users may not be aware of the relationships between them. In addition, dynamic query interfaces require tight integration with the back-end database to allow the smooth scrolling along dimensions. They are therefore not as appropriate for web applications.

6. Future work

Several important directions for future work have already been discussed including the incorporation of magnitude into critiques, the end-user customization of global similarity metrics, and the integration of content-based recommendation using semantic ratings. Several other areas of future work are relevant for the requirements gathering function in electronic commerce.

6.1. Catalog Analysis

In addition to using ratings to analyze and classify users of FindMe systems, we can also use semantic ratings to classify the products based on user reaction. We have only performed some preliminary analyses in this direction, but some of the results are promising. If we can create a profile for each restaurant in the Entree database based on user reaction, we may be able to cluster the restaurants by how users react to them. One

possible purpose would be to enable catalog owners to visualize the catalog contents and determine products that are heavily critiqued in particular ways. With the Entree data, for example, we might want to know what restaurants are considered over-priced or boring.

Another possible type of analysis is “catalog gap” analysis. It is possible using the critique technique to navigate out of the product space entirely, looking for more of some feature than is available in any known product. Such an action is instructive to the user in finding the boundaries of the product space. These dead ends should also be instructive to the catalog owner because they indicate places in the product space where there is demand (or at least interest) and no known supply. For example, if an owner of an audio components catalog sees that users commonly run out of product space when looking for higher-quality components, it would be a good indication that there are high-end audiophiles among the clientele who would be well served with the addition of some top-of-the-line components.

In summarizing user ratings for catalog analysis, we are assuming that there will be consistency in user reactions. Of course, part of the aim of FindMe systems is to not expect a high degree of consistency between a user’s sessions with the system – a diner may want something fancy tonight and something basic next week. However, in aggregate, we are hoping to look for emergent patterns that rise above the noise of individual preference shifts. There is some indication that such consistency can be found. In early experiments with long-term profile data, it appears that users do display consistency in their critiques over time.³

6.2. Multiple constraint types

As described in section 3.3 above, the FindMe technique does not accumulate a short-term profile letting each navigation action limit all future retrievals, since our experience demonstrated that users found such implementations frustrating. However, it is also the case that some constraints are hard constraints that must be satisfied – a cat owner needs an apartment that will accept her pets, a traveler may need to leave on a particular date. The next generation implementation of the FindMe architecture will represent both hard and soft constraints in its API, permitting exploration of the use of both. The simplest approach would be to suggest such constraints based on the critique history, and allow the user to turn them on as desired.

7. Conclusion

How can an e-commerce site connect users with products? This basic task is one that today’s sites do not perform well. Keyword search, the most widely implemented mechanism, has many usability problems, some of which are overcome by next-generation knowledge-based search systems that have emerged from case-based reasoning research. These technologies have reached maturity and are beginning to be deployed commercially. Still, approaches that put requirements first, followed by search,

³ These preliminary experiments repeated the collaborative filtering experiments described above using long-term profiles instead of short-term sessions. The results show only a small decrease in accuracy when long-term profiles were used.

fail to support the full cycle of buying behavior. We propose instead an interactive navigation of the product space, which has the effect of leading the user towards the right product without an explicit “requirements gathering” step. Our approach is a better fit to psychological models of buying behavior in that it supports exploration and feedback between needs analysis and product discovery. In addition, the data generated implicitly by critique-based navigation is much richer than other implicit measures, containing the reasons behind a preference. Our experiments have shown that these semantically-rich ratings can add power to knowledge-weak techniques such as collaborative filtering, and we are beginning to explore possible applications in catalog analysis as well.

Acknowledgements

The original FindMe research was conducted at the University of Chicago in collaboration with Kristian Hammond, with the support of the Office of Naval Research under grant F49620-88-D-0058. The interface to the Entree system was designed and created by Robin Hunicke. Many others contributed to the FindMe effort at the University of Chicago, including Terrence Asselin, Kai Martin, Kass Schmitt, Robb Thomas, and Ben Young. The technology was further developed by Verb, Inc. under a technical team led by Dan Crow.

References

Bergmann, R., Breen, S., Göker, M., Manago, M., and Wess, S. (1999) *Developing Industrial Case-Based Reasoning Applications: The INRECA Methodology*. Lecture Notes in Artificial Intelligence, LNAI 1612. Berlin: Springer-Verlag.

Burke, R. (1999) The Wasabi Personal Shopper: A Case-Based Recommender System. In *Proceedings the 11th Annual Conference on Innovative Applications of Artificial Intelligence*. Menlo Park, CA: AAAI Press.

Burke, R. (2000) Knowledge-based Recommender Systems. In A. Kent (ed.): *Encyclopedia of Library and Information Systems*. Vol. 69, Supplement 32.

Burke, R. (in press) Hybrid Recommender Systems: Survey and Experiments. *User Modeling and User Adapted Interaction*.

Burke, R., Hammond, K., and Young, B. (1997) The FindMe Approach to Assisted Browsing. *IEEE Expert*, 12(4): 32-40.

Cutting, D. R., Pederson, J. O., Charger, D., and Turkey, J. W. (1992) Scatter/Gather: A cluster-based approach to browsing large document collections. In *Proceedings of the 15th Annual International ACM/SIGIR Conference*, pp. 318-329. New York: ACM Press.

Ferguson, A. and Bridge, D. (2000) Partial Orders and Indifference Relations: Being Purposefully Vague in Case-Based Retrieval. In E. Blanzieri and L. Portinale (Eds.), *Advances in Case-Based Reasoning (EWCBR-00), Lecture Notes in AI 1898*, pp. 74-85. Berlin: Springer-Verlag.

Ferguson, W., Bareiss, R., Birnbaum, L. and Osgood, R. (1992) ASK Systems: An approach to the realization of story-based teachers. *Journal of the Learning Sciences*, 2(1), 95-134.

Comment: Need page numbers from AAAI proceedings.

Foskett, D. J. (1980) Thesaurus. In A. Kent, H. Lancour, and J. E. Daily (Eds.), *Encyclopedia of Library and Information Science*, Vol. 30. pp. 416-462. New York: Marcel Dekker. Reprinted in K. Sparck-Jones and P. Willett (Eds.), *Readings in Information Retrieval*, pp. 111-134. San Francisco: Morgan Kaufmann.

Goldberg, D., Nichols, D., Oki, B. M., Terry, D. (1992) Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12), 61-70.

Guttman, R. H. (1998) *Merchant Differentiation through Integrative Negotiation in Agent-mediated Electronic Commerce*. Master's Thesis, School of Architecture and Planning, Program in Media Arts and Sciences, Massachusetts Institute of Technology.

Hagen, P. R., Manning H. and Paul, Y. (2000) Must Search Stink? *The Forrester Report*. June 2000.

Hill, W.C., Stead, L., Rosenstein, M. and Furnas, G. *Recommending and evaluating choices in a virtual community of use*, in Proceedings of CHI'95 (Denver CO, May 1995), ACM Press, 194-201.

Keeney, R. L. and Raiffa, H. (1993) *Decisions with Multiple Objectives*. Cambridge, UK: Cambridge University Press.

Kolodner, J. (1993) *Case-based reasoning*. San Mateo, CA: Morgan Kaufmann.

Konstan, J., Miller, B., Maltz, D., Herlocker, J., Gordon, L., and Riedl, J. (1997) GroupLens: Applying Collaborative Filtering to Usenet News. *Communications of the ACM*, 40(3), 77-87.

Maes, P., Guttman, R. H., and Moukas, A. G. (1999) Agents that buy and sell. *Communications of the ACM*, 42(3), 81-91.

Mullen, B. and Johnson, C. (1990) *The Psychology of Consumer Behavior*. Hillsdale, N.J.: Lawrence Erlbaum Assoc.

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J. (1994) GroupLens: an open architecture for collaborative filtering of netnews. In *CSCW '94: Proceedings of the conference on Computer supported cooperative work*, pp. 175-186. New York: ACM Press.

Riesbeck, C., and Schank, R. C. (1989) *Inside Case-Based Reasoning*. Hillsdale, NJ: Lawrence Erlbaum.

Salton, G., and McGill, M. (1983) *Introduction to modern information retrieval*. New York: McGraw-Hill.

Schafer, J. B., Konstan, J. and Riedl, J. (1999) Recommender Systems in E-Commerce. In *EC '99: Proceedings of the First ACM Conference on Electronic Commerce*, Denver, CO, pp. 158-166.

Schank, R.C., and Riesbeck, C. (1981) *Inside Computer Understanding: Five Programs with Miniatures*. Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Schneiderman, B. (1994) Dynamic Queries for Visual Information Seeking. *IEEE Software*, 11(6), 70-77.

Shardanand, U. and Maes, P. (1995) Social information filtering algorithms for automating "word of mouth" In *CHI-95: Conference proceedings on Human factors in computing systems*, pp. 210-217. New York: ACM Press.

Smyth, B and McClave, P. (2001) Similarity vs. Diversity. In D.W. Aha and I. Watson (Eds.), *Case-Based Reasoning Research and Development (ICCB-01), Lecture Notes in AI 2080*, pp. 347-361. Berlin: Springer-Verlag.

Burke, Interactive Critiquing...

Vollrath, I., Wilke, W. and Bergmann, R. (1998) Case-based reasoning support for online catalog sales. *IEEE Internet Computing*, 2(4): 47-54.

Watson, I. (1997) *Applying Case-Based Reasoning: Techniques for Enterprise Systems*. San Francisco: Morgan Kaufmann.