Mobile Product Browsing using Bayesian Retrieval

Christoph Lofi, Christian Nieke, Wolf-Tilo Balke
Institute for Information Systems
Technische Universität Braunschweig
Braunschweig, Germany
{lofi, nieke, balke}@ifis.cs.tu-bs.de

Abstract— **Reacting to technological advances in the domain of mobile devices, many traditionally desktop-bound applications now are ready to make the transition into the mobile world. Especially mobile shopping applications promise a large potential for commercial. However, in order to work on the limited screen estate even of modern devices, traditional category-based browsing approaches to online shopping have to be rethought. In this paper we design an innovative approach to intuitively guide users through product databases based on Bayesian probability modeling for navigational purposes. Our navigation model is focused on feedback and inspired by content-based retrieval techniques. Moreover, we exploit new features of today's devices like touch screens to ease interaction. Due to the novel interface-related simplicity, our system supports users in their decision process while demanding only minimal cognitive load. We outline the theoretical foundations and the design space of such a system and evaluate its retrieval effectiveness using real-world data sets. In fact, we show that using our probabilistic navigation model about 98% of all searches can be completed successfully with an average of only 3 rounds of feedback on the 4th displayed screen.**

Keywords: mobile e-commerce; mobile interfaces; probabilistic retrieval

I. Introduction

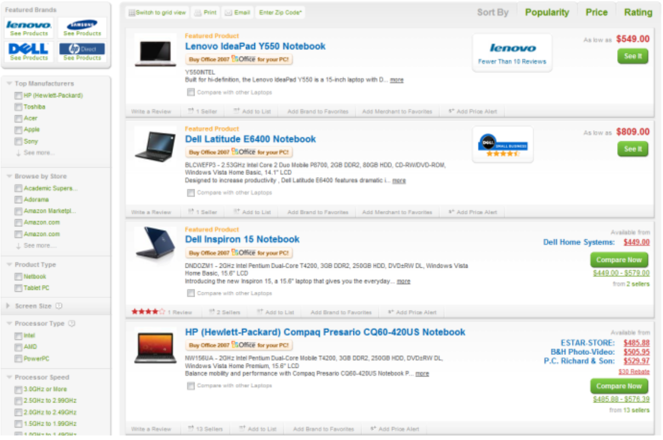
In recent years, mobile services gained tremendous momentum. Encouraged by continuous hardware advances and service cost reduction, these services changed the way mobile devices are used and now range from simple data services, over location- or presence-based services, up to full mobile mesh-up applications. In short, mobile devices are now capable of performing many tasks which were usually in the domain of desktop computing only. However, some usage scenarios are still problematic despite advances in mobile hardware and software design. Especially the limited display size and resolution together with the lack of efficient text input devices hamper the direct adoption of Web and desktop applications.

Mobile shopping applications are a good example for a problematic field in the domain of mobile services. They provide a huge market potential [*1*] [*2*], but this potential is still not fully tapped. This is because until recently mobile e-shopping was mostly performed by specialized Web pages offering a shop's products, e.g., using the Wireless Application Protocol (WAP) [*3*] or i-mode [*4*]. These approaches are, however, not able to deliver the *rich user experience* that is expected from online shopping. In particular, the actual mobile content (either created manually, or provided by adaptive transformations e.g., templating or XSLT [*5*]) is optimized for display, but unfortunately not for user interaction. In contrast, newer models of smart phones are already able to display online shopping Web pages including more (especially navigational) features. But the reduced screen size still obstructs a smooth workflow with a lot of scrolling. Thus the exploration of larger data collections is prohibited.

Consider the scenario of buying a laptop computer. Using a price comparison Website (http://www.pricegrabber.com), we found that over 1,000 models are currently available in the marketplace, each characterized by 19 different attributes. Browsing through such a variety requires complex Web interfaces as well as a considerable amount of screen estate (see, for example, Figure 1). Even state of the art mobile devices will struggle with displaying such sites in a visually manageable manner, and most sites opt for minimal mobile-optimized Web sites for accessing their content (see Figure 2), compromising especially on browsing and filtering interface elements. However, this *increases* the cognitive load on the mobile user as the complexity of the underlying browsing paradigm and data model remains the same while, at the same time, visual representation and navigational features are crippled.

We argue that it is not sufficient to just adapt the presentation of Web data. Instead, it requires a *true mobile workflow* respecting the special characteristics of devices and users to propel a broad-scale success of mobile e-commerce. It is necessary to part with traditional form-based approaches classifying items into complex multi-attribute hierarchies as known from the World Wide Web. Instead, specialized and adaptive interaction paradigms have to be optimized for small screens without the necessity of extensive amounts of scrolling.

In this paper, we present an alternative approach to traditional category- and form-based browsing inspired by content-based retrieval techniques [*6*]. As in query-by-example approaches [*7*,*8*], the user directly manipulates sample items from the database by providing feedback in a straight-forward fashion similar to example-critiquing techniques [*9*]. To select better items for display, we build on top of a probabilistic relevance model pioneered by [*10*] which represents the uncertainty about the users goal as a probability estimate. While our content-based approach requires only a minimal amount of screen estate, it provides effective guidance for a personalized exploration of the underlying data space. Item selection, rele-



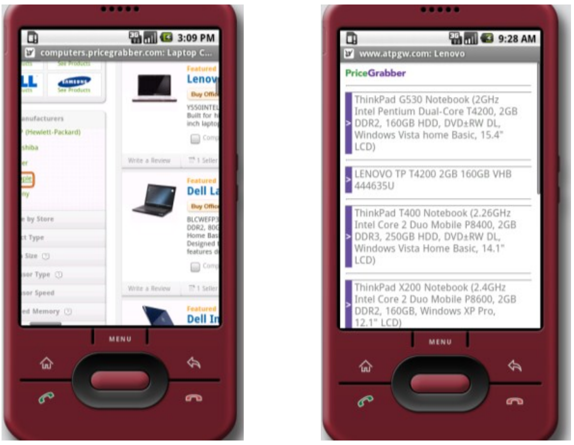
Featured Product
Lenovo IdeaPad Y550 Notebook
Buy Office 2007 Office for your PC!
Y550INTEL
Built for hi-definition, the Lenovo IdeaPad Y550 is a 15-inch laptop with D... more
□ Compare with other Laptops
lenovo Fewer Than 10 Reviews
As low as $549.00
See It
Write a Review | 1 Seller | Add to List | Add Brand to Favorites | Add Merchant to Favorites | Add Price Alert

Featured Product
Dell Latitude E6400 Notebook
Buy Office 2007 Office for your PC!
BLCWEFP3 - 2.53GHz Intel Core 2 Duo Mobile P8700, 2GB DDR2, 80GB HDD, CD-RW/DVD-ROM, Windows Vista Home Basic, 14.1" LCD
Designed to increase productivity , Dell Latitude E6400 features dramatic i... more
□ Compare with other Laptops
As low as $809.00
See It
Write a Review | 1 Seller | Add to List | Add Brand to Favorites | Add Merchant to Favorites | Add Price Alert

Featured Product
Dell Inspiron 15 Notebook
Buy Office 2007 Office for your PC!
DNDOZM1 - 2GHz Intel Pentium Dual-Core T4200, 3GB DDR2, 250GB HDD, DVD±RW DL, Windows Vista Home Basic, 15.6" LCD
Introducing the new Inspiron 15, a 15.6" laptop that gives you the everyday... more
□ Compare with other Laptops
Available from
Dell Home Systems: $449.00
Compare Now
$449.00 - $579.00
from 2 sellers
★★★★☆ 1 Review | 2 Sellers | Add to List | Add Brand to Favorites | Add Price Alert

HP (Hewlett-Packard) Compaq Presario CQ60-420US Notebook
Buy Office 2007 Office for your PC!
NW156UA - 2GHz Intel Pentium Dual-Core Mobile T4200, 3GB DDR2, 250GB HDD, DVD±RW DL, Windows Vista Home Premium, 15.6" LCD
Balance mobility and performance with Compaq Presario CQ60-420US Notebook P... more
□ Compare with other Laptops
Available from
ESTAR-STORE: $485.88
B&H Photo-Video: $505.95
P.C. Richard & Son: $529.97
$30 Rebate
Compare Now
$485.88 - $576.39
from 13 sellers
Write a Review | 13 Sellers | Add to List | Add Brand to Favorites | Add Price Alert

Figure 1. State of the art online shopping browsing (http://www.pricegrabber.com)



ThinkPad G530 Notebook (2GHz Intel Pentium Dual-Core T4200, 2GB DDR2, 160GB HDD, DVD±RW DL, Windows Vista home Basic, 15.4" LCD)

LENOVO TP T4200 2GB 160GB VHB 444635U

ThinkPad T400 Notebook (2.26GHz Intel Core 2 Duo Mobile P8400, 2GB DDR3, 250GB HDD, DVD±RW DL, Windows Vista Home Basic, 14.1" LCD)

ThinkPad X200 Notebook (2.4GHz Intel Core 2 Duo Mobile P8600, 2GB DDR2, 160GB, Windows XP Pro, 12.1" LCD)

Figure 2. Browsing unoptimzed Websites (left)
Browsing specialized mobile sites (right)

vance feedback, and user model adaptation form a *feedback cycle*. After just a few feedback cycles, the system will finally converge to a near-optimal selection of objects corresponding to the user's preferences. The advantages of our approach can be summarized as follows:

- *Clean, easy, and space-efficient user interface*: our approach does not rely on filtering and restricting database items with respect to their attribute values, thus traditional navigation elements can be omitted from the user interface. Furthermore, only a small selection of objects needs to be displayed in an intuitive touch-based feedback interface at the same time.
- *User guidance*: the system will be able to guide the user to his/her preferred database items without the need of extensive domain knowledge (e.g. knowledge of which brands or models are suitable, or sensible combinations of attribute values, etc). Thus, even the exploration of large data sets is easily possible.
- *Low cognitive load*: users are neither required to manually review dozens of product offers on mobile devices, nor to provide exact categorization and attribute values of desired result objects. Instead, he/she only needs to provide simple 'good/bad' feedback on selected examples by simply touching desired features. Thus, the cognitive load of the retrieval process remains low promoting a positive user experience.

II. RELATED WORK

The characteristics of mobile devices significantly differ from desktop computers often used for e-commerce applications. But recently bandwidths have severely improved and also the computing power of today's devices is usually sufficient for display purposes or payment transfers. Still, the predominant limitations are the *small screen-size* and the *inconvenience of inputting information*.

This is reflected by the preferred methods of user interaction for information access: whereas desktop sessions mostly include keyword-searches, a recent study reports that about 94% of all mobile sessions are performed by *browsing navigation* [*11*]. Indeed, although desktop users often use search engines like e.g., Google or Yahoo!, as general portals to information, the study clearly shows that mobile keyword search sessions on average take 3 times longer than browsing sessions and are thus only used in unavoidable cases. A complementary result had been reported earlier by Google specifically investigating mobile search behavior in a large scale study [*12*]: mobile keyword searches are more restricted in the number of query terms (on average 2.3 terms per query), as well as in the number of query refinements users are prepared to perform.

The problem of visualizing a plethora of information on small displays has since long attracted research. Though simple result lists enumerating 20-50 items are commonplace in desktop application interfaces, this obviously cannot be used efficiently on small screens. Still, often a *hierarchical structure* of the content can be exploited. This is particularly true when working with product catalogs in e-shopping scenarios. Recent work can be roughly divided into two approaches: on one hand the content could be segmented into semantically meaningful and coherent parts before adapting it for display, e.g., [*13*] or [*14*], on the other hand novel visualization approaches try to fit more information on the screen without increasing the user's cognitive load, e.g., using radial edgeless trees in [*15*].

Also inputting information for mobile applications has seen a lot of innovation recently. Since typing on miniature keyboards or text input via number panels is cumbersome, today's applications rely on advanced hardware features. For example [*16*] uses the *cell phone's camera* and retrieves Web sites related to photographed products and extracted brand names. Moreover, small, yet accurate *touch-screens* using projected capacitive touch technology like for instance in Apple's iPhone, Motorola's Droid or Palm's Pre, allow for easy input. In both cases the user may restrict him-/herself to a non-verbal style of interaction very similar to the query-by-example paradigm championed in multimedia databases (see e.g., [*7*,*8*]).

In terms of *efficient information exchange* in human-computer interaction, the individual steering of search processes generally rely on feedback mechanisms. The earliest algorithms have already been researched in the 1970ies in the field of information retrieval. Nevertheless, many basic features (like the shifting of query term weights) can still be found at the heart of today's retrieval systems on the Web. In the area of product searches on the Web the work on *example-critiquing* in [*9*] is of major importance in relation to our approach. Here, users with a rather vague understanding of the domain and available products can be guided towards best-matching products by criticizing unwanted aspects of samples. By integrating this feedback into a ranking function, user studies show that users have a positive interaction experience: they feel involved in the choice process and therefore tend to like the final results better. However, the accurate tuning of ranking functions for individual users still remains unsolved.

Another related technique that was recently developed for mobile information retrieval is given in [*17*]. Focusing on a scenario of document retrieval (newspaper articles) the benefits of using individual feedback even on only a small fraction of retrieval results are clearly shown. In particular, the authors show how basic user feedback can be turned into a *decision tree* representing all possible outcomes of the user interaction with the system. In contrast to our work, however, inferior selection strategies are used and an explicit user model based on Hidden Markov Models has to be constructed.

III. Mobile Shopping Using Feedback on Examples

Before focusing on the theoretical aspects, let us consider two short usage scenarios for our approach. Assume that Frank is interested in buying a new laptop computer. Frank thus visits a local electronics retail store. While browsing the showcased offers, he discovers a *netbook computer* which catches his interest. He particularly likes the *form-factor* of the device, while the *performance specifications* leave room for improvement. As he is not particularly familiar with netbooks and the store only offers a small selection, he requests decision assistance from a shopping portal using his smartphone. After entering the netbooks product name, its specifics are displayed.

Let us assume that the netbook Frank discovered features a 9” screen, weighs 950g, and offers 512 MB RAM and an older Celeron M processor. Besides the product’s specifications, the shopping system may also provide a selection of typical values for similar products with respect to major attributes in a separate feedback area, e.g., for screen sizes the values 7”, 9”, 10”, and 11” could be displayed. By tapping on the respective values in the feedback area, Frank states that he likes the size and weight of the currently displayed product, but would rather prefer a larger main memory and a better processor. Incorporating this feedback into the next selection step the system shows Frank a product better satisfying his preferences. After Frank has reviewed just a few netbooks presented by the shopping portal, he gains a better understanding of the netbook domain and thus can settle on a certain model in an informed way.

Or assume a different customer Cathy looking for a new notebook computer while commuting to work. Cathy might have no particular brand preference, thus she initializes her mobile shopping portal by simply selecting the notebook category. Her session starts with displaying the currently most popular notebook which may be a powerful desktop replacement machine. In the feedback area, a wide range of other attributes characteristics is displayed, spanning from those usually found in netbooks to those in the desktop replacement category. Although Cathy might have only vague preferences, she feels that the currently displayed notebook is too heavy to be easily portable and provides respective feedback for a more lightweight machine. In the next interaction step, she is presented a lighter notebook with an adapted selection of feedback possibilities. After browsing through several notebooks, Cathy may finally settle on a mid-sized business laptop.

IV. Theoretical Foundations

Our theoretical foundations are based on Bayesian retrieval approaches like suggested in [*10*]. Such approaches are quite often used in multimedia databases, but need significant modifications to be employed in general retrieval scenarios. To be self-contained, we first briefly summarize the basic theory of Bayesian retrieval and then focus on necessary modifications and possible design decisions for optimizing the approach to support navigation in a mobile shopping scenario.

As stated above the basic interaction with our system consists of a *feedback cycle*. In each step $t = 1, 2, \ldots$ during a user interaction, the user will be shown a set of *selected database objects* D_t. After reviewing the objects, he/she will provide feedback on the selection’s items in form of a so-called *user action* A_t. A user action might rate each item of set D_t, simply select good or bad examples, or offer a more sophisticated attribute critique. When discussing the algorithmic design space in the next chapter, we will introduce several models for user actions. An ordered collection of displayed database items and the respective user actions up to a certain interaction step t is called *interaction history* and denoted as:

$$H_t = A_1,\ D_1,\ A_2,\ D_2,\ \ldots\ ,\ A_t,\ D_t.$$

The history at any given step t can be used to derive a new selection D_{t+1} for the next feedback cycle $t + 1$, using different *selection strategies,* which will be discussed in the next chapter. All strategies rely on analyzing the probability estimates for each database item representing the belief that this particular item is indeed the one the user is looking for (given the current history). We denote database items by $T_1, T_2, \ldots, T_n$. Each object T_i is annotated with a probability of being the user's target T. We will denote the initial *a-priori estimate* before starting the user interaction by $P(T = T_i)$. Again, there are several strategies for providing a *startup distribution* of the a-priori estimates (which will be discussed later). After each interaction step, the probabilities are updated to *a-posteriori estimates* respecting the current interaction history. This is denoted by $P(T = T_i \mid H_t)$, i.e. the probability of T_i being the users target T considering the current interaction history H_t. Using Bayes' rule the equation for computing the new estimates is:

$$\boldsymbol{P(T = T_i \mid H_t)} = \frac{P(H_t \mid T=T_i)\, P(T=T_i)}{P(H_t)} = \frac{P(H_t \mid T=T_i)\, P(T=T_i)}{\sum_{j=1}^{n} P(H_t \mid T=T_j)\, P(T=T_j)}. \quad \textbf{(1)}$$

Note that the probability for a given history $P(H_t)$ can be represented by summarizing for all items T_i in the database the respective a-posteriori probability that this particular object is indeed the users target T combined with the objects’ a-priori probability, i.e. $P(H_t) = \sum_{j=1}^{n} P(H_t \mid T = T_j)\, P(T = T_j)$. The a-priori probability $P(T = T_j)$ can directly be extracted from the startup distribution. As the user interaction is performed in multiple feedback cycles, the above equation can also be transformed into a recursive form using only the current user action and the history of the previous cycle: $P(T = T_i | H_t) = P(T = T_i | D_t, A_t, H_{t-1})$. Since D_t is deterministically given by the selection strategy and H_{t-1} is always known, this results in equation 2:

$$P(T = T_i \mid H_t) = \boldsymbol{P(T = T_i | D_t, A_t, H_{t-1})}$$
$$= \frac{P(D_t, A_t \mid T = T_i, H_{t-1})\, P(D_t, T = T_i \mid H_{t-1})}{\sum_{j=1}^{n} P(D_t, A_t | T = T_j, H_{t-1})\, P(D_t, T = T_j | H_{t-1})}$$
$$= \frac{\boldsymbol{P(A_t | T = T_i, D_t, H_{t-1})}\, P(T = T_i \mid H_{t-1})}{\sum_{j=1}^{n} \boldsymbol{P(A_t | T = T_j, D_t, H_{t-1})}\, P(T = T_j | H_{t-1})}. \quad \mathbf{(2)}$$

The term $P(T = T_j | H_{t-1})$ can be computed recursively until the a priori approximation given by the start distribution is reached. But the central term within the previous equitation $P(A_t \mid T = T_i, D_t, H_{t-1})$ remains difficult, i.e. the probability that the user will actually perform the current action A_t given that the current database object T_i is indeed the target T and given the previous history H_{t-1}. This term essentially predicts a user's action considering all currently known information and thus resembles a *user model*. Selecting suitable user models is a considerable challenge when designing an implementation. We discuss suitable user models in the next chapter.

V. Design Space

In this section, we focus on the design space and design decisions for actually implementing a system suitable for mobile devices. We discuss the problem of startup distributions, the object selection strategy, choices for user models, and finally approaches for user interfaces for navigational access.

A. Startup Distributions

The startup distribution gives the a-priori probability $P(T = T_i)$ for each database object T_i of being the target object T before the user starts interacting. Thus, the startup distribution addresses the current sessions' cold start problem. Several approaches for solving the problem can be identified.

The *naïve approach* is assuming an uniform a priori distribution, i.e. $P(T = T_i) = {}^{1}\!/_{n}$ for each $T_i \in \{T_1, \dots, T_n\}$. This does not require any additional information and does not impose any bias on the data. However, the number of required feedback cycles can be reduced by incorporating general heuristics satisfying a larger user base. For example, a-priori probabilities could be chosen such that they are dependent on the items' popularity with respect to e.g., previous system sessions or sales numbers. By incorporating popularity measures into the initial setup, especially interaction sessions for users with vague preferences can be supported.

In contrast, the user interaction may also directly be initialized with a *positive example product*. For this approach, the user directly provides an object which is set to a high a-priori probability. Using any distribution model (e.g. a normal distribution), similar objects also gain a higher a-priori probability decreasing with the distance to the selected item. Thus, interaction may start with a high initial focus, but still allows exploring interesting alternatives in the surrounding feature space.

B. User Models and the Feature Space

As seen in chapter IV, the probability of each object being the target can be calculated using equation 2, however, requiring a user model to evaluate the term $P(A_t | T = T_j, D_t, H_{t-1})$, which represents the probability of a specific user action given a target, a selection of objects, and a history. In [*10*] several approaches to create user models are discussed featuring the *item-based feedback* paradigm, i.e. users provide binary feedback on complete items ("I like/don't like this product"); *weighted* and *attribute-based* user models will be introduced later in this section.

We assume user decisions to be time-invariant, i.e. given a set of displayed objects and a target, the user always takes the same actions, independent of the history. This allows simplifying the term for the user model: $P(A_t | T = T_j, D_t)$. We further assume that users make decisions to choose each of the displayed items $D_t = \{X_1, \dots, X_{N_D}\}$ independently. Following [*10*] we also take a softmin approach to calculate the probability for each decision on each single object X_a as follows:

$$P_{soft}\big(A = a \big| X_1, \dots, X_{N_D}, T_j\big) = \frac{\exp(-d(X_a, T)/\sigma)}{\sum_{i=1}^{N_D} \exp(-d(X_i, T)/\sigma)} \quad \mathbf{(4)}$$

In this formula, $d(X, T)$ denotes the distance between item X and the target T, while σ is a factor to modify the 'softness' of the function (set to 1 in our case). The softmin yields highest values for those items X closest to target T, decreasing with the distance. It can be interpreted as probability of an equal variant Gaussian with mean X_a to create the target T. Using the assumption of independent decisions, a combined decision can be calculated by multiplying the probabilities of each single decision, e.g., $P_{soft}(A = 1, 2) = P_{soft}(A = 1) * P_{soft}(A = 2)$.

While formula 4 defines the user model, it does, however, still miss a definition for the distance $d(X, T)$ of two objects. This leads us to the *feature space*, the representation of real world objects, i.e. the laptops, in the system. Basically each object is represented as an n-tuple X_i of features. In the simplest case, features are *numerical* and represented by their actual values, e.g., disk space or CPU frequency. Another type are *Boolean* features like Bluetooth support, which are either present or not. Unfortunately, *categorical* features (e.g. brand or color) are more difficult and are best represented as ranked score values either encoding the user's preferences or some common preference (e.g. performance benchmark results).

In the following, we assume that each vector X_i contains only numerical values, i.e. the actual value for *numerical features*, 0/1 for *Boolean features* or the rank of a *categorical feature*. In order to ensure a fair and unbiased comparison of individual attributes, each attribute value is normalized to an interval $[0, 1]$, with 0 representing the lowest attribute value and 1 the highest value. For some attributes, logarithmic normalization can be used (e.g. main memory size for notebook computers which is usually exponential to the base 2). The distance between two database objects can finally be defined by the Euclidian distance measure of their respective feature vectors. However, this simple distance function does not respect varying importance of certain attributes, e.g. a user might deem CPU performance as being more important than a laptop's weight.

Consider the following example: some item shows a feature f_p perfectly meeting the users' expectations, another well matching feature f_g, and three other features f_1, f_2, f_3 which are far off. However, using the current model, the user can only provide feedback for the whole item. If he/she does not provide positive feedback for the current object, he/she misses the chance to tell the system about the preference for f_p and f_g. However, if he/she does provide positive feedback for the object, this could have undesired effects as the system implicitly also assumes that the features f_1, f_2, f_3 are preferred by the user. This leads to two problems: a high cognitive load for the user who needs to decide whether or not positive feedback for a whole object will benefit the search, and "noise" within the retrieval by the undesired features f_1, f_2, f_3.

To remedy this problem, the user has to be enabled to weight individual attribute values of displayed items according to his/her preferences. In our implementation users can highlight each attribute value of a display object to be *very good*, *good* or *not interesting*, which leads to numerical weights of 2, 1 or 0 respectively. We then use a vector of weights w_a for each feature of item *a* to create a weight matrix $W_a = w_a * E$. Of course the weights have to be included in the user model:

$$P_{soft}\left(A = a \middle| X_1, \dots, X_{N_D}, T_j, w_a\right)$$

$$= \frac{\exp(-d(X_a W_a, T W_a)/\sigma)}{\sum_{i=1}^{N_D} \exp(-d(X_i W_a, T W_a)/\sigma)} \quad \mathbf{(5)}$$

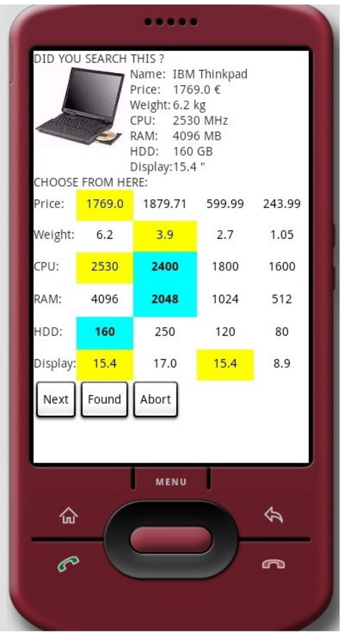
Note that this weighting is only used for a single user action and the respective recalculation of probabilities; for the next interaction cycle new weights can be defined.

This model may also be extended with additional information on the semantic distance of objects. For example, [*18*] proposes an approach for mining semantic similarities of object objects by combining user sentiments found in online discussion forums or review Websites with pure attribute values as given by the database objects. Thus, also the perceived distance of two objects can be respected.

C. Selection Strategies

The selection strategy's task is to select for each feedback cycle t_i a suitable subset of the database D_i to be displayed. There are two basic strategies: the most-probable and the most-informative selection strategy. Our experiments showed an attribute-weighted hybrid approach to be most suitable.

The *most-probable strategy* was first proposed in [*19*]. After determining the optimal number of objects N_i required for the next selection D_i, the N_i objects with highest a-posteriori probability $P(T = T_i | H_t)$ are returned, breaking ties arbitrarily. While this strategy performs satisfiable, it is highly susceptible to overfitting. Especially in early phases or with uniform startup distributions, the algorithm tends to gravitate to particular clusters within the data depending on the initial selections and early user feedback. After gaining momentum, this trend is only hard to stop by the user because only items are offered for feedback that will further bias selections towards the initial direction. An example: In the first step of the feedback cycle, the user might get a selection only of cheap products. If the user chooses any, this will boost the *a-posteriori probability* of all cheap products. But then the next selection of most probable items will again contain only cheap items, too.


DID YOU SEARCH THIS ?

Name: IBM Thinkpad
Price: 1769.0 €
Weight: 6.2 kg
CPU: 2530 MHz
RAM: 4096 MB
HDD: 160 GB
Display: 15.4 "

CHOOSE FROM HERE:

Price:	1769.0	1879.71	599.99	243.99
Weight:	6.2	3.9	2.7	1.05
CPU:	2530	**2400**	1800	1600
RAM:	4096	**2048**	1024	512
HDD:	**160**	250	120	80
Display:	15.4	17.0	15.4	8.9

Figure 3. Interface for relevance feedback

In contrast, the *most-informative strategy* [*10*] aims at reducing the number of required iterations while avoiding dead-ends caused by overfitting. This is achieved by focusing the current feedback selection D_i on objects such that feedback on these objects rewards the greatest information gain possible while also varying the selection. Basically, this approach aims at minimizing the projected number of remaining iterations by evaluating all possible selections, based on the current distribution $P(T = T_i)$. To approximate the number of required user feedback steps, denoted as $C[P(T)]$, the concept of entropy from information theory is used. For resolving the ambiguity given by $P(T = T_i)$, the following entropy notion emerges:

$$C[P(T)] \approx -a \sum_{i=1}^{n} P(T = T_i) \log\left(P(T = T_i)\right) \quad (6)$$

While the most-informative strategy can indeed reduce the number of feedback cycles and deadlocks in some settings, it introduces a new set of problems: firstly, minimizing entropy over all possible selections is computationally very expensive, which calls for an approximation technique like e.g. Monte-Carlo sampling. The quality of an optimization on a sample however is strongly depending on the quality of the random sample itself. If the sample does not already contain informative tuples the information gain will be low, decreasing performance. Even worse: assuming the feedback already created a high probability for the target, if the target is not part of the sample, it will not be presented to the user, thus the search cannot be completed. Another problem is of psychological nature: as sample objects are not necessarily chosen to be probable but to lead to a high information gain, users are often confused as they are presented objects seemingly unrelated to the feedback previously provided, which might lead to premature interaction cancellation.

To counter the disadvantages of both approaches, we propose a hybrid strategy combing their individual strength. Instead of treating all N_i objects of the current selection D_t as being equal, we emphasize a small subset as being the systems current target suggestion and display them in a detailed fashion. The remaining items are only used for feedback-related purposes. By choosing the emphasized objects using the most-probable strategy, it is ensured that users perceive the interaction process as smoothly converging. On the other hand, in case a user is not yet satisfied, he/she is offered to use the feedback-objects for further steering. These objects are selected using the most-informative strategy; thus, high convergence speed can be maintained and overfitting is avoided.

D. User Interface

Designing an interface for a mobile application has critical requirements, in particular applications have to be *easy to handle* and *space efficient* regarding screen estate. To comply with these goals, our prototype uses only a minimal amount of dedicated control elements. We implemented the prototype using the Google Android SDK version 1.5. As displayed in Figure 3, the upper half of the screen always shows detailed information about the current most probable object. In the lower half, the system displays all of the current selectable objects, one per column, and allows users to select a weight for each attribute feature by simply clicking on the desired attribute. A single click selects the feature as *good* (yellow background), a second click selects the feature as *very good* (bold with cyan background) and a third click deselects the feature.

The only dedicated control elements are the buttons on the bottom of the screen. Note that this application can be fully controlled via touch interface, which makes it *easy to handle*, while almost all the control elements are also displaying information at the same time, which makes it *space efficient*. In our case, the number of displayed objects affects the efficiency of the algorithms, which leads to the following problem: to achieve a high information gain in each iteration step, a large number of objects should be displayed to the user. This, however, increases the cognitive load, and requires screen space. Since psychological results in [*20*] suggest that a human cannot efficiently handle more than four pieces of information at a time, we decided to present only four objects on the screen.

VI. Evaluations

In this section, we extensively evaluate our proposed method using various scenarios and interaction models. Our evaluation approach is twofold: first, we perform automated simulations on different configurations of the algorithmic design space. Secondly, we perform a user study on a selected subset of the configurations used in the preceding simulations. In both cases, we use a representative real world dataset containing 1,350 notebook offers which were crawled from the Web.

A. Implemented Prototypes

In order to gain a better understanding of the complex design space, we implemented a modular prototype application which allows testing of different combinations of approaches. The following configurations of content-based retrieval algorithms were implemented, using a uniform a priori distribution as startup distribution:

- *MP*: In this configuration, four database objects are presented to the user as the current selection D_t using the *most-probable* selection strategy and a user model according to formula 4. During tests, however, we found this configuration to easily run into dead ends.
- *MI*: To counter the problems in version *MP*, we exchanged the selection strategy to the *most informative* scheme. While this version avoids dead ends, it misses feedback to the user, as discussed in section V.C.
- *H*: This configuration employs a hybrid approach to avoid the problems encountered with the former versions. Four database objects are presented to the user in each cycle, one is selected using the most-probable strategy, which gives the user a feedback on the current status of the search, while the other three objects are selected using the most-informative strategy, thus avoiding dead ends.
- *HAttr*: As discussed in section V.C, we found a purely object-based user model to lack means of explaining the user's choices. Therefore we used our extended attribute-weighting user model according to formula 5, which allows the user to weight each attribute of each object as *very good*, *good* or *not interesting*. The selection strategy for the objects is still a hybrid approach.

B. Simulation

In order to reliably establish performance baselines for all former discussed configurations, we employ extensive simulations to obtain large numbers of statistics of synthetic user interaction sessions. Each simulation starts with a uniform startup distribution before emulating a user interaction with the system. To realize such simulations, proximity-based target searches are performed, i.e. for each simulation run a user target T is randomly selected from the database which represents the current user's preferences. The simulated interaction ends successfully if a database object is selected by the system for presentation showing the same or closely similar attribute values (as measured by an Euclidian norm) to the user's target T. After each successful simulation run, the number of presented screens is recorded, representing the convergence performance of this particular run. As our experiments with human users state that a run is aborted unsuccessfully after an average of 7.2 screens, the simulations abort a run as unsuccessful after 15 screens, assuming that the simulation will perform worse than a human.

The simulations chose objects for positive feedback based on a *Converging Threshold* Paradigm. The simulation will choose every object for positive feedback which has a shorter distance to the target than a given threshold. After each step, the threshold will be updated to the shortest distance encountered so far, to simulate a user's rising expectations. For configuration *HAttr*, an attribute is selected as *good* if it is below the threshold and *very good* if it is a perfect fit. A simpler *Best Choice* paradigm, which selected the one best fitting tuple, was tested as well, but showed almost identical results.

C. User Study

To ensure the results of our simulations are actually comparable to real world scenarios, we performed additional tests with human users. Since we wanted to test (and validate) our algorithms' convergence behavior rather than the general usability of mobile devices, our test group consisted of several computer science students (i.e. the test group was already familiar with the use of mobile devices and the general domain of laptops). Each user got a short explanation of the user interface, including which of the displayed objects were *most probable* and which were *most informative*. Similar to the simulated user sessions, for each run the user was then displayed a randomly picked object from the database as his intended target. If a matching object was found, the run was terminated successfully. If the user did no longer expect the target to be found in the near future (at the latest after 15 interaction steps), the run was terminated unsuccessfully. All users quickly understood the concepts and were immediately able to use the mobile interface.

D. Results

The results of the simulations and human tests are shown in Figure 4. The most probable (MP) configuration (first chart) showed acceptable performance in simulations, featuring a success rate of roughly 80% and a average number of screens of 7. During human evaluations, the success rate lowers to just 41% with an average screen number of 4.9. This is mainly due to the fact that the human testers lost patience with the system as they did not perceive significant improvements of the object selection during the individual feedback steps and aborted the interaction session.

The simulation results for the most-informative configuration (MI) can be found in the second chart. This configuration shows significantly lower performance in simulations than the previous most-probable setup (success rate of just 17% with 8.3 screens on average). The reasons for this low performance were already mentioned in V.C: using sampling reduces accuracy, and the most probable target is not displayed, which further prolongs the search. However, human test subjects were able to deal with the most-informative configuration in a much better way, increasing the success rate up to 52% with 5.8 screens in average and thus archiving better results as previously seen with the most-probable strategy.

To remedy the shortcoming of the previous setups, a hybrid selection strategy (H) was employed. This resulted in a success rate of 43% with 8.9 screens in average for simulations. While the hybrid strategy performed worse than a pure most-probable strategy during simulations, human testers were able to achieve good results using this approach; and as shown in the third chart a success rate of 74% with 4.8 screens on average was achieved. The one most-probable object of the selection screen provided a good summary of the current assumed target while

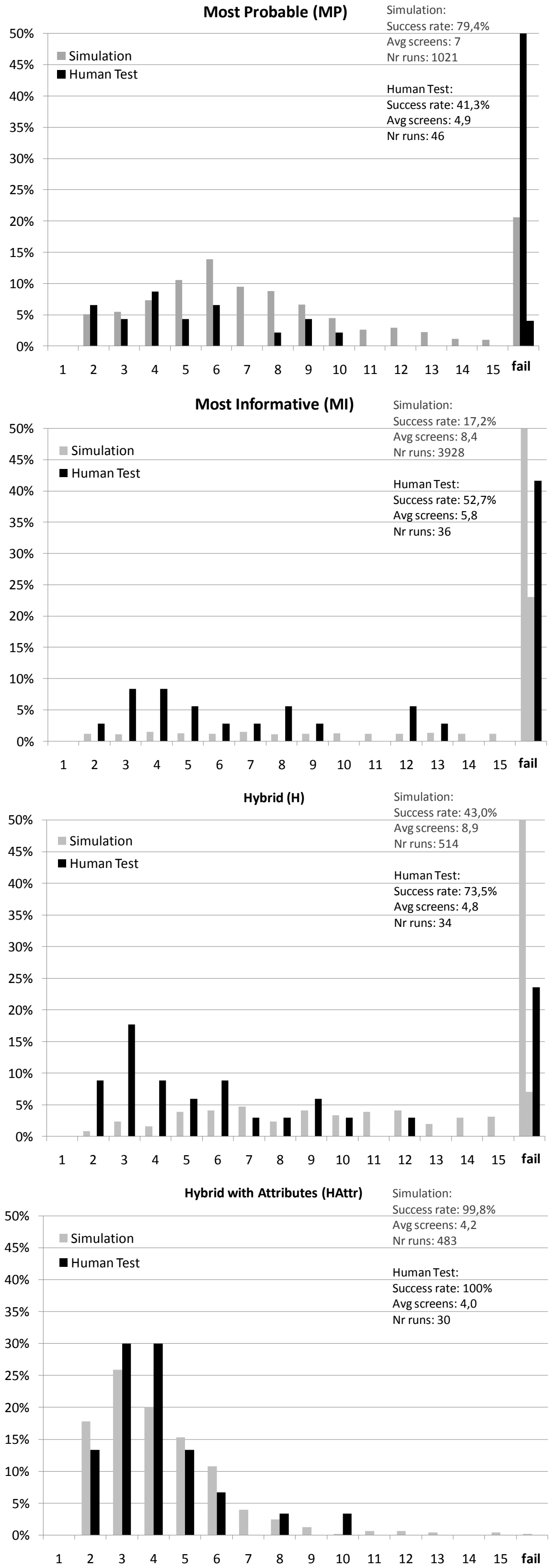


Figure 4. Simulation Results for the four algorithm configurations using simulations and human tests. Columns show distribution of convergence speed (x-axis, represented by number of displayed selection screens until success) across the simulation runs. Last column represents failed runs.

the remaining objects, derived with the most-informative strategy, provide enough variability to successfully steer the election process.

Finally, we evaluated our proposed selection and feedback strategy based on a hybrid selection strategy with weighted-attribute feedback (HAttr). During simulations, this approach showed excellent performance with a success rate of 98% and an average of 4.2 screens (fourth chart). Furthermore, this strong result could be affirmed by the human evaluations with a success rate of 100% and a similar average of 4.0 screens. Thus, this attribute-based strategy showed superior performance compared to any object-based approach. With its high convergence speed, high success rate, and clean user interface this approach is clearly suited for usage in mobile applications.

VII. Summary and Outlook

Following the huge technological advances in the domain of mobile devices, many traditionally desktop-bound applications now are ready to make the transition into the mobile world. In this paper, we argued that by exploiting recently available and easy to use interfaces like touch screens together with an improved user centered navigation model, even more complex applications like mobile shopping can meet the challenge and provide a satisfactory user experience.

As a proof of concept we presented space efficient, easy-to-use content-based retrieval approach for mobile shopping scenarios. In contrast to traditional categorical querying and retrieval processes, our approach especially tailors to the requirements and limitations of mobile devices. Basically, the user is presented with a representative example item from the database and may provide intuitive feedback on desired improvements with respect to the current example's attributes using only a simple touch interface. This feedback is then incorporated into the system by generating and modifying an underlying probability distribution of the users expected most preferred items. Following approaches from the area of multimedia retrieval we used a Bayesian probabilistic user model. This model indeed allowed for feedback-driven user guidance such that the user finds suitable products matching his/her preferences even for large datasets after just a few and simple feedback cycles. We also found that this kind of interaction allowed users to better understand the underlying domain and clarify their preferences during the interaction session.

In our experimentation we used a practical data set extracted from an online laptop retail store. Exploring the algorithmic design space we evaluated different configurations of our approach using synthetic user simulations, as well as real user evaluations. During our experiments, we could clearly show that our hybrid selection strategy combined with weighted attribute value feedback showed superior performance compared to other configurations. On average, the user's desired database object was reliably returned after just three interaction steps on the fourth screen. This performance still has potential for improvement by further optimizing the underlying selection strategies and data models by, e.g. incorporating semantic information and distance measurements or selection strategies additionally respecting psychosocial aspects.

References

[1] Jeffrey L. Funk, "The future of the mobile phone Internet: an analysis of technological trajectories and lead users in the Japanese market," *Technology in Society*, vol. 27, no. 1, 2005.

[2] Larry Freed, "Mobile Apps: The Next Big Thing," Foresee Result, 2009.

[3] Wireless Application Protocol Forum, *Official Wireless Application Protocol: The Complete Standard*, John Wiley & Sons, Ed., 1999.

[4] NTT DoCoMo Japan, "i-Mode Specification," 2001.

[5] Matthias Wagner, Wolf-Tilo Balke, and Werner Kießling, "An XML-Based Multimedia Middleware for Mobile Online Auctions," *Enterprise Information Systems*, vol. III, 2002.

[6] A. Yoshitaka and T. Lchikawa, "A survey on content-based retrieval for multimedia databases," *IEEE Transactions on Knowledge and Data Engineering*, vol. 11, 1999.

[7] T. Kato, T. Kurita, N. Otsu, and K. Hirata, "A sketch retrieval method for full color image database-query by visual example," *Pattern Recognition*, 1992.

[8] Carlton W. Niblack et al., "QBIC project: querying images by content, using color, texture, and shape," in *Storage and Retrieval for Image and Video Databases*, San Jose, CA, USA, 1993.

[9] Paolo Viappiani, Boi Faltings, and Pearl Pu, "Preference-based Search using Example-Critiquing with Suggestions," *Journal on Artificial Intelligence Research (JAIR)*, vol. 27, 2006.

[10] I.J. Cox, M.L. Miller, T.P. Minka, T.V. Papathomas, and P.N. Yianilos, "The Bayesian Image Retrieval System, PicHunter: Theory, Implementation, and Psychophysical Experiments," *IEEE Transactions on Image Processing*, vol. 9, no. 1, 2000.

[11] Karen Church, Barry Smyth, Paul Cotter, and Keith Bradley, "Mobile Information Access: A Study of Emerging Search Behavior on the Mobile Internet," *ACM Transactions on the Web (TWEB)*, vol. 1, no. 1, 2007.

[12] Maryam Kamvar and Shumeet Baluja, "A Large Scale Study of Wireless Search Behavior: Google Mobile Search," in *SIGCHI Conference on Human Factors in Computing Systems*, Montréal, Québec, Canada, 2006.

[13] Yevgen Borodin, Jalal Mahmud, and I.V. Ramakrishnan, "Context browsing with mobiles - when less is more," in *Int. Conf. On Mobile Systems, Applications And Services*, San Juan, Puerto Rico , 2007.

[14] Hamed Ahmadi and Jun Kong, "Efficient web browsing on small screens," in *Conf. on Advanced Visual Interfaces*, Napoli, Italy, 2008.

[15] Jie Hao, Kang Zhang, Chad Allen Gabrysch, and Qiaoming Zhu, "Managing Hierarchical Information on Small Screens," in *J. Int. Conf. on Advances in Data and Web Management*, Suzhou, China , 2009.

[16] Yan Xu and Mirjana Spasojevic, "Designing a Vision-based Mobile Interface for In-store Shopping; Jiang Gao; Matthias Jacob," in *Nordic Conf. on Human-Computer Interaction*, Lund, Sweden , 2005.

[17] Vishwa Vinay, Ingemar J. Cox, Natasa Milic-Frayling, and Ken Wood, "Can constrained relevance feedback and display strategies help users retrieve items on mobile devices?," *Information Retrieval*, vol. 9, no. 4, 2006.

[18] Jongwuk Lee, Seung-won Hwang, Zaiqing Nie, and Ji-Rong Wen, "Query result clustering for object-level search," in *Int. Conf. on Knowledge Discovery and Data Mining*, Paris, France , 2009.

[19] I.J. Cox, M.L. Miller, S.M. Omohundro, and P.N. Yianilos, "Pichunter: Bayesian relevance feedback for image retrieval," in *Int. Conf. on Pattern Recognition*, Vienna, Austria, p. 1996.

[20] D. Benyon, P. Turner, and S. Turne, "Designing Interactive Systems," 2005.

[21] Paolo Viappiani, Boi Faltings, and Perl Pu, "Preference-based Search using Example-Critiquing with Suggestions," *Journal of Artificial Intelligence Research*, vol. 27, 2006.