Information Seeking: Convergence of Search, Recommendations and Advertising

Hector Garcia-Molina   Georgia Koutrika   Aditya Parameswaran

1. INTRODUCTION

All of us are faced with a "deluge of data" [2], in our workplaces and our homes: an ever-growing World Wide Web, digital books and magazines, photographs, blogs, tweets, e-mails, databases, activity logs, sensor streams, on-line videos, movies and music, and so on. Thus, one of the fundamental problems in Computer Science has become even more critical today: how to identify objects *satisfying a user's information need*. The goal is to present to the user only information that is of interest and relevance, at the right place and time.

At least three types of *information providing mechanisms* have been developed over the years to satisfy user information needs:

- A *search* mechanism takes as input a query that describes the current user interests. A body of objects (e.g., documents, records) is searched, and ones that somehow match the query are returned. For example, in a Web search engine, a user may enter a sequence of words (the query), and is given a ranked list of Web pages that contain the desired words. In a database system, the query is typically more structured (e.g., I want the names of products with price less than 100 dollars) and the search is over structured records.

- A *recommendation* mechanism typically does not use an explicit query but rather analyzes the user context (e.g., what the user has recently purchased or read), and if available, a user profile (e.g., the user likes mystery novels). Then the recommendation mechanism presents to the user one or more descriptions of objects (e.g., books, people, movies) that may be of interest. For example, in an electronic commerce site, when a user purchases one object, he may be shown a set of similar objects, or objects that other people have purchased together with the just purchased object. In a Web search engine, when a user types in one query, he may be shown a list of related queries that others have entered following the first query.

- An *advertisement* mechanism is similar to a recommendation mechanism, except that the objects presented to the user are commercial advertisements, and financial considerations play a central role in ranking. An advertising mechanism also considers the user context, e.g., what Web site is currently being visited, or what query the user just entered. The selection of advertisements can be based on the similarity of the context to the ad, or the similarity of the context to a bid phrase (where the advertiser specifies the contexts he is interested in), or on financial considerations (e.g., which advertiser is willing to pay more for the ad placement).

The main thesis of this paper is that these mechanisms are not that different to begin with, and designing these three mechanisms making use of this commonality could lead to significant benefits.

All three mechanisms share the same common goal (Figure 1): matching a context (which may or may not include an explicit query) to a collection of information objects (ads, product descriptions, web pages, etc). The way context can be described varies, but there is no fundamental reason why one object type needs to be associated with one mechanism type. Similarly, the way matches are determined and ranked varies, but again, there is no fundamental reason why approaches cannot be used across mechanisms.

There are of course differences in the way information needs can be met, but these differences are often orthogonal to the mechanisms and the way they have been used traditionally. To be more specific, consider Table 1 which illustrates the different features information providing mechanisms may have (leftmost column, one approach per row) and the choices typically made by mechanisms (column headers).

The first row, for instance, refers to the delivery mode: pull vs. push. With information pull, the user makes a specific request for information, e.g., submits a query or clicks on a request button (e.g., get information on product X). With information push, the user is given information he did not specifically request. For example, when a user visits the home page for Bermuda, he is shown a banner ad at the top of the page for a hotel in Bermuda. As we discussed above, traditional search mechanisms are pull. However, the distinction becomes blurred if we consider standing queries that are run periodically (e.g., show me the weather report for my city every morning), or if the search mechanisms can use the context to predict queries that may be of interest to the user. Similarly, ad mechanisms are push because the user does not explicitly request advertisements. But

why shouldn't a user be allowed to explicitly search the ads database? Recommendation mechanisms have been used in the past in either pull or push mode. With pull, recommendations are explicitly requested (e.g., what movie should I rent), but push recommendations are shown when the user did not ask for them (e.g., users who bought this book also bought that book).

The row labeled "Beneficiary" refers to whether the interests of the user or the information provider have priority. With traditional advertisements, the main priority is the provider, e.g., priority is on ads that make money as opposed to ads that are of interest to the user (but the user is unlikely to buy something). With traditional search, the results are typically not biased due to financial considerations. But again, the distinction can be blurred, and has been blurred in the past. For example, an early search engine (`goto.com`, which later merged with Yahoo!) mixed Web search results with paid advertisements, so was essentially providing a search mechanism that benefited both the user and the provider. Even today, the maps website `maps.google.com` provides a service where businesses can get "preferred" listings (for a fee) that are shown alongside ordinary listings. However, most search engines today prefer to clearly separate "paid" or "sponsored" results from unbiased results by using different areas of the results page for each. We predict that in the future the distinction will be blurred, as it is already done in other areas (e.g., television "infomercials", fan web sites or blogs for celebrities or new products, advertisements appearing with Twitter posts, etc.)

The row labeled "Unexpected good" refers to whether "unexpected" (serendipitous, or novel) results are beneficial to the user. For example, say a user is searching for "exchange rate euro dollar". The user may just be looking for that one number because he is going on a trip and is estimating his expenses. In this case, other information such as the exchange rate for other currencies or the current price of gold, are of little interest. However, if the user is an investor who was not thinking of investments in precious metals, the unexpected news that "gold is selling at a low price today" may actually be of interest. Of course, it is hard to tell what the user may find interesting, but recommendation systems have explored ways to find those unexpected but useful bits of information. On the other hand, traditional search mechanisms (and to some extent ad mechanisms) focus more on obtaining information that is clearly relevant to the user's current query or activity. But again, this distinction is not necessary, and if the context indicates that a user is in exploration mode, then it may be appropriate to show some unexpected results.

"Collective knowledge" in Table 1 means that the context of other users (e.g., queries or purchase history) is used. In general, collective knowledge can improve results, but is not always exploited. "Query available" means that the user explicitly described his information need. For instance, as mentioned above, an advertising mechanism may select ads to display based on a user query (query available) or simply based on a web page the user is viewing (query not available). "Context dependent" means that the user's context is taken into account.

The goal of this discussion-style paper is to briefly review for the non-specialist some of the key concepts that historically evolved for search, recommendations and adver-
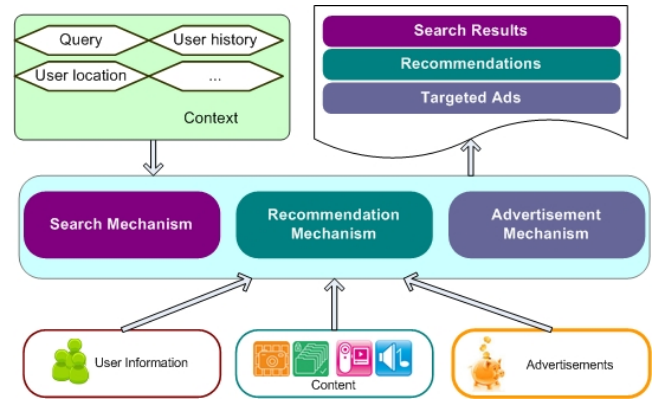

Figure 1: Unifying Search, Recommendations and Ads.

tisement. Then, we provide our opinion on why we believe a convergence of these technologies would be more useful today than in the past. Looking ahead, we also discuss some of the challenges standing in the way of this convergence. Finally, we conclude by presenting a "case study" we have focused on at Stanford, and how we have tried to address some of these research challenges.

Note that achieving convergence at established organizations with huge existing investments in search, recommendations and advertising may be difficult in the near future. However, a startup that is building a system from scratch, might find convergence much more attractive and feasible, and might get a competitive advance from doing things differently. Thus, we predict that we will see more efforts at combination of technologies from small companies or university projects, rather than from the established players that want to protect their existing investments.

	Search	Recommendations	Ads
Delivery Mode	Pull	Push or Pull	Push
Beneficiary	User	User and provider	Provider
Unexpected good?	No	Yes	No
Collective knowledge	Maybe	Maybe	Yes
Query available	Yes	Maybe	Maybe
Context dependent	Maybe	Maybe	Maybe

Table 1: Differences in Information Providing Mechanisms

2. BRIEF RETROSPECTIVE

2.1 Search

Search is the most mature of the three technologies we review in this section, and several excellent textbooks exist on the subject (for example, [21, 9]). At the core of any search mechanism is a *search engine* or a *query processor* that finds and ranks a set of objects (or documents) based on a user query. Conceptually, there are two separate stages for the search process: a *filtering* stage and a *ranking* stage. (Note we are describing a conceptual way to think about search; actual systems may or may not implement these stages separately.)

The input to the filtering stage is a set of objects (the corpus) and a filtering or Boolean sub-query, while the output

is the subset of the objects that satisfy the sub-query. Thus, the filtering sub-query is a predicate that examines the attributes (or properties) of one object and evaluates to true or false. For example, "price is 100 and brand is Nikon" and "object's title contains words Brazil or Argentina and object is a pdf document" are two predicates. There are many ways to express filtering queries, and different levels of "expressive power," but we do not survey all the options here.

The input to the ranking stage is a subset of the objects and a ranking sub-query, while the output is a ranked version of the input objects. In general, each object is given a score indicating how well that object matches the sub-query. As a very simple example, if the sub-query is "Brazil Argentina," the score of an object may be the total number of times the words (terms) Brazil and Argentina occur in it. The more the words occur, the more relevant the object is judged to the sub-query and hence may be given a higher score. More sophisticated scores such as TF-IDF [21] also consider the relative size of the object (e.g., document length), as well as the relative importance of each keyword in the sub-query. In addition, we may also consider scores denoting the inherent goodness of the objects independent of the query, such as pagerank or hub/authority score.

Beyond these basics, there are many variations and improvements, with roughly two goals in mind: reducing latency and improving the relevance of the results to the query. Since huge numbers of queries need to be processed against huge numbers of objects, run time latency is a critical issue. One important data structure, often used to reduce latency, is an inverted-list index. For each possible word, such an index gives the list of all objects where the word occurs. These lists can then be manipulated to answer queries. For instance, if the filtering query asks for objects that have the words Brazil and Argentina, we can fetch the inverted lists for each of these words, intersect the lists, and quickly get a list of the matching objects. The inverted-list index can be partitioned across multiple computers in order to reduce latency. Caching is another powerful tool that can help: the results for common queries, or frequently used inverted lists can be kept in memory to speed up searches.

The scoring function is supposed to rank higher objects that better match the user's interest: the better this matching, the more relevant the search. There are many ways to improve this relevance; here we just mention a few examples. The list of words in the ranking sub-query can be automatically expanded to include related words. For instance, if the sub-query is for "car", we can convert it to "car automobile". After a user receives results, he can indicate which of the objects were of interest (relevance feedback). The system can then automatically reformulate the query and present new results that presumably will better match the user interests. The system can also pre-process the original objects to identify words that are more important. For example, if the objects are clustered, the words that best describe each cluster can be considered important and given a higher weighting when computing scores.

Finally, note that the query-driven search process we have described here is often complemented by a navigation style search. Typically, a user poses a query and examines the objects at the top of the ranked list. Those objects may contain links or references to other objects, and the user may navigate to those. While strictly speaking those user actions are not part of the search, if additional software (such as a toolbar) is installed, the search engine can track that navigation and assist in the navigation. For example, the search engine can use the visited objects as context, in order to modify its ranking function of future searches. Or the search engine can generate additional links or tags (words that can be used in future searches), to complement the links and words already in the observed objects, to help the user reach the desired information.

2.2 Recommendations

Recommendation mechanisms provide advice on objects (e.g., movies, products, travel and leisure activities) depending on the user context or profile (and not so much on an existing query) [5]. They can be broadly classified by the *strategy* they employ (content-based or collaborative filtering), and by the *recipient* of the recommendations (individual user or group recommendations).

Recommendation strategy. *Content-based filtering* recommends objects similar to those the user liked in the past by analyzing the associations between the user's past choices and the descriptions of new objects [11, 27]. While content-based filtering does not take into account actions of other users, *collaborative filtering* analyzes past actions and behavior of all users. Collaborative filtering identifies interesting associations between users or between the objects that the users have seen or purchased which can be used to make recommendations to a single person. Note that collaborative filtering may have difficulty with "cold starts", i.e., making recommendations for new objects or new users for which there is no prior history. *Memory-based* collaborative filtering algorithms employ statistical techniques to identify users similar to the target user, and then recommend objects based on the preferences of the similar users [13, 16, 28]. *Model-based* approaches use past user behavior to build off-line models (e.g., Bayesian [14], probabilistic relational [15] and probabilistic LSA models [17]) that capture relationships or patterns between users or objects. These models are subsequently used to compute recommendations on-line.

Another example of a collaborative approach is that of the discovery of sets of products usually purchased together by many independent buyers, which is applied to recommend to a shopper other products related to the product currently being viewed (as in the "Customers who bought this object also bought" recommendations in Amazon [1]).

Recommendation recipient. Recommendations may serve a single person or a group of people. A group can be formed on a *recurring* basis, e.g., friends who meet regularly for dinner, or on an *ephemeral* basis, e.g., random users getting together for a weekend climbing trip organized by their sports club [6]. Regardless how the group is formed, the semantics of *group recommendations* are significantly different in comparison to individual user recommendations.

In individual user recommendation, the usual goal is to retrieve objects with the highest scores computed by a given recommendation strategy. For example, Netflix (`netflix.com`) recommends movies that are more likely to be rented by a user by finding those with the highest expected rating by that user. In group recommendation, however, an object may have different relevance to different group members and this disagreement among members must be resolved. The job of the group recommender is to make suggestions that reflect the preferences of the group as a whole, while offering reasonable and acceptable compromises to individual group

members. There are different strategies for combining individual user preferences to adapt to groups, some of which are inspired by Social Choice Theory [22]. For instance, a user's preference for an object is treated as a vote and the object with the most votes may be recommended (also called *plurality voting*).

2.3 Ads

The search mechanism is provided free to consumers because search results are coupled with advertisements. *Sponsored search* is the delivery of relevant advertisements (a type of information object) as part of the search experience and it has evolved to satisfy the needs of users for relevant search results and the advertiser's desire for qualified traffic to their websites.

The point of sponsored search is to give a mechanism by which advertisers can get users to visit their websites [18]. An *advertiser* is a person or organization interested in generating user traffic to a particular website for some specific purpose. An *advertisement* or a *sponsored link* is a set of keywords along with the associated URLs, titles, images (also called *display*) and descriptions, to be displayed on a search result page. Sponsored links can also be provided on websites that are deemed relevant to the advertiser's content or on hosted email services. This linking is known as *contextual advertising* [29].

Search advertising is typically sold and delivered on the basis of keywords. Search engines conduct running *auctions* to sell ads according to *bids* received for keywords and the relative relevance of query keywords to ads in the inventory. For example, the keyword 'home mortgage refinancing' is more expensive than one that is in less demand, such as 'population of Alaska'. When a user submits a query, the search engine matches each advertiser's keywords (that describing their ads) to the user query, makes sure that the advertiser's content is relevant to the targeted keyword and selects relevant ads to be displayed along the search engine results for the user query based on the advertiser bids.

When the user reviews the search result pages, and clicks on a sponsored link, the user's browser displays the advertiser's web page pointed to by the link. The search engine tracks this click, along with all the other clicks by this user as well as other users within a given period. At the end of this period, the search engine bills the advertiser, providing various statistics concerning the outcome of the advertiser's campaign. The advertiser may also track the conversion rate, i.e., the fraction of clicks that resulted in a purchase or desired action at the advertiser's site. From an analysis of these statistics, the advertiser can alter bids, maximum budget per period, and selection of keywords in real time. By engaging in the search process and 'buying' search keywords, these advertisers become active participants in the information seeking process of users in a very dynamic fashion. Typically, search advertising activities can be measured in the following ways [3]:

- *Impressions*: When a website displays an ad to a unique user it counts as an *impression*. Websites that display banner ads typically count the number of unique impressions, charging advertisers a CPM (Cost Per thousand viewers) rate.

- *Clicks*: Some search engines count the number of times users click on a particular ad (typically taking the user to the advertisers website), charging advertisers a CPC (Cost Per Click) rate. The fraction of impressions that get clicked on is called the CTR (Click-Through Rate).

- *Actions*: Some search engines charge advertisers based on CPA (Cost Per Action), which tracks the cost for completing specific activities, such as new customer enrollment or making a sale.

3. CONVERGENCE

Historically, the three information providing mechanisms we have reviewed have remained separate because they catered to different needs. Recommendation systems of the past were restricted to a single homogeneous domain and returned a list of movies or books with no keyword search possible. Search engines, on the other hand, were geared towards returning web-pages that satisfy keyword search, with little or no emphasis on personalization or identification of intent. Advertisements were developed independently as a way to monetize search. In most cases the goal was to keep paid advertisements clearly separate from "organic" search results, so it was natural to have separate systems for each.

Another reason for the three mechanisms to have remained independent so far is performance. For instance, in the past, search engines took a lot of processing time to perform intersections of inverted lists and return relevant results. Today, due to a combination of a powerful back-end infrastructure, caching and pre-processing, and better algorithms, search engines return results extremely rapidly, and there is now potential for further improvement in the quality of search results.

Over time, users have started demanding more from their information providers, and as a result we already see more powerful systems that expand beyond their core functionality. For instance, several recommendation sites now support keyword search and filters (e.g., Netflix, Amazon and eBay), and many recommendation sites provide recommendations across different heterogeneous product categories. Due to the abundance of web-pages relating to the same keywords, search engines now focus on ascertaining user intent (instead of performing a vanilla keyword search and ranking), and providing results (i.e., information objects) that are interesting to the user, either based on his past experience or that of other users similar to the target user (e.g., social search sites such as aardvark (`aardvark.com`), hunch (`hunch.com`)). We now see ads, not just for traditional search, but in product recommendation sites, such as Amazon and eBay. Companies have recognized that providing advertisements along with their recommendations (suitably distinguished from the recommendation results) can be extremely profitable. For instance, the auction site ebay (`ebay.com`) provides a *deal of the day* for all visitors to the site, in addition to *buy it now*, special items directly sold from a provider for a fixed price – both of these are essentially advertisements.

Information Consolidation Engine: We believe that both users and system builders will benefit from a generalized Information Consolidation Engine (ICE). Such an ICE should present a single interface to the user for all information needs, and should share technologies across search, recommendation and advertisement.

The unified interface should allow users to specify their needs and interests, whether it is through traditional search

boxes, filters, examples of objects that were previously of interest, or friends with similar needs or interests. (Of course, the interface should also allow negatives, e.g., I am not interested in Harry Potter books). The universe of information objects to consider through this one interface should be as broad as possible, e.g., should cover Web pages, tweets, blogs, photos, movies, products, etc. Clearly, such a unified interface would be beneficial, as users would not need to search differently depending on what they want, and they would obtain a much richer set of information objects. There is of course the danger of information overload, e.g., if we give the user photos of Harry Potter when she was thinking of books. Thus, inferring user intent will be even more critical than it is already on a more narrow search interface.

At the backend, an ICE should merge the functionality provided by search, recommendation and advertising. For instance, in all three cases there is the need to filter information objects (including ads) based on user input. The objects that meet the filtering constraint, also need to be ranked. In all cases, the search and ranking needs to be driven by user interests, which should be captured with generalized notions of personalization and context. Thus, if a user is not interested in Harry Potter, he should not be shown web pages about that fictional character, he should not be given recommendations for Harry Potter books and movies, and he should not be shown related advertisements. Similarly, if the user's friends like iPhones, then search results, recommendations and advertisements should be appropriately biased.

Advantages for a unified ICE: A unified ICE would not only modularize the design of individual components (e.g., filtering, ranking, personalization), but would also allow a reuse of components for each of the information providing mechanisms. This reuse would come with obvious benefits: improvements in one component affects all providing mechanisms and designing new providing mechanisms would simply be a matter of specifying which combination of components need to be used.

While we are arguing that a unified ICE is desirable, and will lead to a cleaner system design and more powerful functionality, we are not saying it will be easy to implement. The convergence and the desire to give users more powerful functionality leads to interesting challenges, some of which we discuss in the next section.

4. DIRECTIONS AND CHALLENGES

Search, recommendation and advertising technologies are being extended and improved along several dimensions. We briefly describe some of these directions, emphasizing that the directions are very often shared across the technologies.

4.1 Complex Targets

From the universe of objects (blogs, books, ads, web pages, places, movies, ...), the information consolidation engine selects some objects that satisfy the user's information need.

In most traditional applications for search, recommendation and advertising, there is a single object that is being sought, such as when the user is looking to rent a movie from Netflix, or picking a restaurant to have dinner.

However, there are information needs that require a "set" or a "package" being shown to a user instead of a single object. For instance, say a person visits a customizable computer hardware store. In this case we may recommend a

printer, a keyboard, a monitor and so on, as a "package" [30]. In a movie store, we may want to recommend sets of movies that have extra value if watched or purchased together. For instance, the site `amazon.com` suggests books that can be purchased together for a discount. Note that ranking packages is different than simply ranking the individual items: with packages one must take into account the synergy provided by combining items. In the computer hardware example, the provided set involves objects that are from different categories while the movies and books examples had all objects belonging to the same category (movies or books). Thus, sets can be *homogeneous* or *heterogeneous*.

Multidimensional recommendations (or search results) [4] is an instance where we provide a *heterogeneous* package. In this case, each dimension corresponds to a different kind of object. For instance, we might suggest that the user watches a romantic film (dimension 1) with his spouse (dimension 2) at home (dimension 3), or that the user watches a comedy film (dimension 1) with friends (dimension 2) at the cinema (dimension 3).

Another example of a set satisfying information needs is that of travel packages. In this case not only are we providing objects from different categories (flights, hotels), but we are also suggesting a *consumption plan*, i.e., telling the user that she must first "consume" the part of the trip corresponding to Las Vegas before "consuming" the part corresponding to Los Angeles. When recommending courses to students in a university, we need to recommend a consumption plan in addition to the courses, especially since not every course is offered in every quarter.

4.2 Constraints

Constraints provide a way for users to specify what they want from their information providers. Constraints also include physical or institutional restrictions that one needs to take into account while providing information objects. Constraints fall into one of four categories: (a) Filter (b) Package (c) Sequencing (d) Prohibitive.

Filter constraints let the user specify to the information consolidation engine how to narrow the space of recommendable objects. Filter constraints were introduced in Section 2.1 as part of traditional search. However, they can also be used for recommendations or ads. For example, a user may indicate that he only wants recommendations for movies in English or that he does not wish to see any ads targeted at senior citizens. In all cases, filter constraints apply to a single object from the set (movie or ad in our examples above).

Package constraints lets the user specify a numerical constraint on the entire package of items being provided. One common type of package constraint is a budget constraint, i.e., the budget (money, time, units) that a user can utilize on the set of objects being provided. For example at the computer hardware store, the customer can specify that he can spend a maximum of $2000. At the movie store, the customer can specify that he wants to rent 5 movies. This constraint could also be related to time, for example, in course recommendations for an undergraduate student, we have 4 years as the total time budget. The package restriction could also be institutionally-imposed, for example, some events do not allow users to purchase more than a certain number of tickets at the same time. In course recommendations, department-specified graduation requirements specify

a negative constraint (e.g., the student must take *at least* 3 math classes). In search advertisements, there are a limited number of slots that can be used, and that effectively acts as a budget. In addition, the search engine needs to pick the right ads such that they retain users while maximizing search revenues.

While suggesting a consumption plan, *Sequencing* constraints specify that objects need to be consumed in a certain order (could be a partial order). For instance, in course recommendations, we cannot recommend a course to a student if he/she has not satisfied its prerequisites. In a TV video streaming site (like Hulu `hulu.com`), we would recommend users watch the first episodes of a show before watching latter episodes.

Prohibitive constraints specify what pairs, triples, etc. of objects in a set cannot be provided together. For instance, a travel plan with two flights with not enough time to make the connection should not be suggested, two courses that have roughly the same content should not be taken, and so on.

4.3 Personalization

The concept of personalization [23] is simple: the more the information consolidation engine knows about a *user's profile*, i.e., person's history, consumer patterns, past searches, and interests as an individual as well the *user's context*, that is her current location, device, search goals, the better position it is in to provide better, customized search results, recommendations or ads. Here are examples of personalization:

Personalized recommendations: When a user logs on, Amazon (`amazon.com`) or Ebay (`ebay.com`) is able to immediately suggest some objects to purchase by tracking previous searches as well as purchasing history.

Personalized Search results: Large web-search engines have been "personalizing" search to some extent for years. Users in the UK will get different results searching for certain terms, especially commercial ones, than users in the USA, and users in San Francisco may get different results from those in New York. In addition to location information, previous searches can also be used to personalize search results. Google (`google.com`) performs search personalization for all users [12].

Personalized ads: Yahoo's advertising service, SmartAds [31], uses behavioral, demographic and geographic information to try to connect people with marketing that caters to their interests. For instance, if a 25-year-old man in San Francisco is shopping online for a car, then he could be shown a banner ad for the Toyota Prius which is a popular car for people in his demographic. The ad could list prices specific to San Francisco and could direct him to his nearest dealer for a test drive. The web site can obtain his age and gender from his `yahoo.com` account information. The search engine identified that he was in the market for a car. His IP address or maybe even his weather channel settings could have alerted the system that he was probably a San Francisco resident.

There are different levels of personalization: from coarse-grained (e.g., based on your country or city of residence) to finer-grained (e.g., personalized ads in sites based on your recent search history). Personalization can be experienced within the bounds of the same system (e.g., search engine, social site, news site, etc.) based on whatever information is locally known about the user. Personalizing information objects based on the user's location or past searches within a search engine are instances of *intra-site personalization*. Personalizing user experience on one site based on information collected and possibly aggregated over different sites is what we call *inter-site personalization*. (There is a gray area when different web sites are operated by the same company.) Early examples of this form of personalization have been observed in the context of ads. For example, a user's recent searches on a search engine may result in related sponsored links appearing on a partner web site. Recently, Facebook (`facebook.com`) launched a small pilot program with an exclusive set of partners, currently the restaurant aggregator site `yelp.com`, Microsoft online document management site `docs.com`, and customizable internet radio site `pandora.com`, to offer a personalized experience to Facebook users. These partners have been given access to public information on Facebook (e.g., names, friend lists and interests and other information users have shared on their Facebook profiles) to personalize a user's experience on the partner's site.

Personalization is not always desirable, and if not done properly may backfire. In the remainder of this section we review some of the factors that can be used in answering the question: "Personalization or not, and to what extent?"

Personalization accuracy. Personalization can be fairly accurate assuming that we have fairly accurate information about a user's interests as well as the objects/topics the user is seeking. Personalization is suitable for "power" users (very active movie watchers, taggers, and so forth). For unknown users, we need to resort back to traditional methods with no personalization [7]. Finally, personalization depends on the type of objects a user is looking for. For example, for some queries (e.g., "US population" or "rachel ray recipes"), many people who issue the query are looking for the same thing. For other queries (e.g., 'famous poems'), different people want very different results even though they express their need in the same way [32].

User information. Personalization can be solely based on information known about a single user (*atomic personalization*). Atomic personalization may sometimes lead to *over-specialization*. Making a recommendation of something a user would definitely like is valuable but at the cost of serendipity and diversity. (See also Sec. 4.4.) Taking into account others' information, such as users who bought similar objects, or friends' interests in a social network can help cope with over-specialization. *Collaborative personalization* can also help when little or no information is known about an individual.

Extent of personalization. Personalizing results based on the user's location or past searches results in a "changed world view": once the engine decides it knows what you want, your world view is narrowed further, and objects are provided from the limited view. For instance, localized search is great when one is searching for restaurants locally but may not be desirable when one just want to find the best restaurants (worldwide). In fact, a non-personalized result may provide interesting and unexpected objects. This would never happen if the information provider never provides results that do not match the past profile. The solution is to dynamically determine when personalized content should be returned and when global quality weighs heavily for a particular user. Bringing the user into the picture may also be

required by allowing him to explicitly control the extent of personalization.

Privacy. Personalization comes with a price: loss of privacy, which can now be experienced at different levels. In intra-site personalization, the more the system knows about the user, the better it can serve the user. If an opt-out option is provided, then the user can choose whether and to what extent the system stores information about the user. Collaborative personalization comes with a host of privacy issues. For instance, which information from friends can be used in providing information objects to a user? If a user starts receiving ads about baby-related products, she may link this to some friend expecting a baby while the latter has not announced it yet. These effects may be more intrusive and less controllable.

4.4 Metrics

Evaluating information consolidation engines can be highly subjective because results can be user, keyword as well as context-specific. For instance, a romantic film may be a good suggestion for a young couple, but might be a poor suggestion for a child. Thus, there is a need for metrics to evaluate information consolidation engines. In addition to evaluation, metrics also provide desiderata that any information consolidation algorithm should aim to achieve. In both cases, which metrics to pick is the central question.

In the past many metrics have been proposed for advertising, search and recommendations. In advertisements, the traditional metrics have been covered in Sec. 2.3, and mostly revolve around measurement of cost and user attention. In search, the traditional metrics that are studied are precision (Do I return only relevant results?) vs. recall (Do I return all the relevant results?). In recommendations, the typical metric that is used is accuracy (Am I able to predict what the user will purchase/rate?).

However, these metrics are not without flaws, and are not sufficient to evaluate recommendations and search. In particular, recall is not important when there are more high-precision objects than the user can find time to peruse, as is the case with a lot of search queries, and it is arguably more important to provide the most "well-rounded" content. Additionally, the accuracy metric in recommendations suffers from the "uncertainty principle", i.e., recommendation systems that are optimized to do well on accuracy do not take into account the fact that if the consumer had been recommended "better" objects in the first place, he might have chosen those objects instead. By measuring the accuracy, we are in fact measuring the ability of the recommendation system to predict what another recommendation system recommended to the user!

There have been other metrics proposed to evaluate recommendations (and any information results in general) from different perspectives. For instance, there are metrics that require a user to provide feedback. However, most of these metrics are hard to define precisely and many of them overlap significantly. We have identified two different "axes" that are central to these metrics. The axes are (a) expectedness (b) goodness. *Expectedness* tells us how expected a given result is to a user. For example, recommending an analysis course to a math major is probably expected, since analysis is a core course for math majors. (Of course, expectedness, just like other metrics, may vary from user to user.) Transparency is a similar notion to expectedness – this is a measure how "explainable" the recommendation is. *Goodness* is a measure of how much the user likes the object or package. Note that goodness is not the same as "predictability", "precision" or "accuracy" — because a user may want to be recommended objects that he might never purchase just because he wants to be aware of such objects. For instance, a user interested in gadgets might be interested in tablet PCs, even though he is never likely to purchase them. Thus, goodness is a highly subjective metric as well.

While the metrics above define how good a single object is, there are metrics that judge a package of objects, e.g., *diversity* (does the set contain objects that are different from each other?), and judge whether recommendations can be made across all users, e.g., *coverage*.

Although choosing which metrics to use to evaluate our results is hard, choosing metrics to optimize from a algorithm perspective is even harder.

Typically, for a given user profile, the information consolidation algorithm needs to use a combination of *personalization*, i.e., picking and choosing objects that the user would like while others may not, as well as *generalization*, i.e., choosing to provide objects that other users have liked. Different users would need different amounts of personalization and generalization. For instance, a novice movie watcher might prefer more generalization (i.e., good, expected and popular movies), while a connoisseur would prefer more personalization (i.e., good, unexpected).

Additionally, different domains require optimization of different metrics. For instance, in course recommendations, recommending expected courses might be more beneficial to the student in order to complete his course requirements rather than novel courses.

4.5 Other directions and challenges

There are other equally significant issues and challenges encountered in search, recommendation and ad mechanisms. *Trust* for example is important both from the system and the user side. The system side involves the explicit calculation and integration of trust and reputation in search results, ads or recommendations. From a user perspective, users may be allowed to explicitly express their web of trust[1] [8], e.g. users they trust about ratings and reviews or web providers for content, and evaluate trust in content received. In both cases, building trust-aware mechanisms is a challenge especially since malicious users often try to exploit these mechanisms to their benefit. For instance, there has to be explicit control on malicious and adversarial ratings on sites such as `yelp.com`, or these ratings could result in incorrect recommendations or search rankings.

Explainability of provided content is important for helping the user understand system behavior and control how the system uses her personal information. It is also important for helping the system understand the user since the latter can potentially provide better feedback based on her better understanding of the content received.

User feedback is essential for improving system behavior. Many sites use mechanisms to incentivize their users to provide explicit feedback in the form of ratings, comments, and so forth. For example, Netflix provides a "rate movies you've seen to discover movies you'll love" feature to encourage peo-

---

[1]PGP [33] was one of first popular systems to explicitly use the term "Web of Trust", though it was not in the context of search or information flow.

ple rate movies in order to get better recommendations. Collecting explicit feedback is a challenge and providing good incentives to users is a big issue. For this reason, other systems rely on implicit feedback, such as clicks, purchases and so forth. For instance, Amazon uses a user's purchase history to improve recommendations. Understanding implicit feedback can be tricky as it does not always provide a positive indicator of user preference.

5. A CASE STUDY: COURSERANK

Over the past few years we have worked on a system called CourseRank, where we observed the natural need for consolidating search and recommendations. Although CourseRank does not currently have advertising, we see it also naturally fitting into a unified framework. We briefly describe our experience with CourseRank as a case study supporting such a consolidation.

CourseRank is social site where Stanford students can review courses and plan their academic program [10]. Using CourseRank, students can search for courses of interest, evaluate courses taken and receive personalized recommendations based on what other students have taken and liked. Faculty can also add comments to their own courses, and can see how their class compares to other classes. CourseRank has been successful in Stanford and has been already adopted by many other US universities because in addition to features common to other social sites, it provides special tools geared to the academic domain, such as a 4-year course planner that helps students structure their courses over multiple years. In addition, unlike other social sites, it provides access to three types of information: (a) official university data, including course descriptions and schedules, (b) personal information on students (e.g., their class, major, the courses they have already taken and their grades) and (c) course evaluations in the forms of ratings and comments.

In CourseRank, we have implemented novel mechanisms based on the rich data we have at our disposal. These services include: a recommendation engine that lets students personalize their recommendations; a requirements service that checks if a student has met their program requirements, and incorporates missing courses into recommendations; and a novel interface that facilitates discovery of new courses.

Grand Challenge: We envision within CourseRank an information consolidation engine, which allows users to give keywords, constraints and even questions, and provides in return books, course descriptions, course schedules, instructors, questions, comments and answers. For instance, for a course-related information need, the engine will provide courses that match the keywords (i.e., search results), those that are rated highly (i.e., recommendations), or simply those that the university wants to publicize, either due to low enrollment, or when the course is new (i.e., advertisements).

Thus, in CourseRank, not only are we proposing that the same techniques be used for search, recommendations and advertisements, we are also proposing that a single interface (with explanations) be used to "unify" the consolidated information.

Current Research: Our research has touched each of the research challenges outlined above. We worked on the problem of formalizing a language for personalization of recommendations [19]. We have made steps towards addressing the problem of set recommendations under a certain subset of constraints (requirements as well as prerequisites) [25, 24]. We have studied how sequencing information can be used to provide consumption plans automatically using user data [26]. We have studied how tags can be used to create a unified interface to browse and receive search results or recommendations [20].

6. CONCLUSIONS

The three information providing mechanisms, search, recommendation and advertising services, are extremely important in today's information age. We have argued that these three services have many similarities and fit naturally under a single roof. And as we strive to improve these services, e.g., by providing personalized information, or by providing information that meets user constraints, we can enhance all services simultaneously.

We outlined the notion of an information consolidation engine, combining components from search, recommendations and advertising. Due to the obvious benefits of this consolidation engine, we pose the design of such an engine as an important and novel challenge to the research community, with implications for all current information providing methods. This design goal has several sub-problems, some of which we have outlined in this paper.

7. REFERENCES

[1] Amazon (2009): url: http://amazon.com/.
[2] The data deluge. The Economist, February 25 2010.
[3] Search advertising metrics: http://en.wikipedia.org/wiki/search_advertising.
[4] G. Adomavicius, R. Sankaranarayanan, S. Sen, and A. Tuzhilin. Incorporating contextual information in recommender systems using a multidimensional approach. *ACM TOIS*, 23(1), 2005.
[5] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. on Knowledge and Data Engineering*, 17(6), 2005.
[6] Sihem Amer-Yahia, Senjuti Basu-Roy, Ashish Chawla, Gautam Das, and Cong Yu. Group recommendation: Semantics and efficiency. In *VLDB*, 2009.
[7] Sihem Amer-Yahia and Georgia Koutrika. Third International Workshop on Personalized Access, Profile Management, and Context Awareness in Databases (PersDB 2009). *SIGMOD Record*, 38(4), 2009.
[8] S. Ba, A. B. Whinston, and H. Zhang. Building trust in online auction markets through an economic incentive mechanism. *Decision Support Systems*, 35(3):273–286, 2002.
[9] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern information retrieval*. ACM Press, 1999.
[10] Benjamin Bercovitz, Filip Kaliszan, Georgia Koutrika, Henry Liou, Aditya Parameswaran, Petros Venetis, Zahra Mohammadi Zadeh, and Hector Garcia-Molina. Social sites research through courserank. *SIGMOD Record*, 38(4), 2009.
[11] D. Billsus and M. Pazzani. User modeling for adaptive news access. *UMUAI*, 10(2-3), 2000.
[12] Official Google Blog. Personalized search for everyone. *http://googleblog.blogspot.com/2009/12/personalized-*

search-for-everyone.html,
2009.

[13] J.S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *14th UAI Conf.*, 1998.

[14] Y.-H. Chien and E.I. George. A bayesian model for collaborative filtering. In *Seventh Int'l Workshop Artificial Intelligence and Statistics*, 1999.

[15] L. Getoor and M. Sahami. Using probabilistic relational models for collaborative filtering. In *WEBKDD*, 1999.

[16] D. Goldberg, D. Nichols, B.M. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. *C. of ACM*, 35(12):61–70, 1992.

[17] T. Hofmann. Latent semantic models for collaborative filtering. *ACM TOIS*, 22(1), 2004.

[18] Bernard J. Jansen and Tracy Mullen. Sponsored search: an overview of the concept, history, and technology. *Int. J. Electronic Business*, 6(2), 2008.

[19] Georgia Koutrika, Benjamin Bercovitz, and Hector Garcia-Molina. Flexrecs: Expressing and combining flexible recommendations. In *SIGMOD Conference*, 2009.

[20] Georgia Koutrika, Z. Mohammadi Zadeh, and Hector Garcia-Molina. Data clouds: Summarizing keyword search results over structured data. In *12th International Conference on Extending Database Technology (EDBT)*, 2009.

[21] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval.* Cambridge University Press, 2008.

[22] J. Masthoff. Group modeling: Selecting a sequence of television items to suit a group of viewers. *User Modeling and User-Adapted Interaction*, 14(1):37–85, 2004.

[23] Alessandro Micarelli, Fabio Gasparetti, Filippo Sciarrone, and Susan Gauch. *The Adaptive Web*, volume 4321 of *LNCS*, chapter Personalized Search on the World Wide Web. 2007.

[24] Aditya Parameswaran, Petros Venetis, and Hector Garcia-Molina. Recommendation systems with complex constraints: A courserank perspective. In *Stanford InfoLab Technical Report, available at http://ilpubs.stanford.edu:8090/909/*, 2009.

[25] Aditya G. Parameswaran, Hector Garcia-Molina, and Jeffrey D. Ullman. Evaluating, combining and generalizing recommendations with prerequisites. In *CIKM*, pages 919–928, 2010.

[26] Aditya G. Parameswaran, Georgia Koutrika, Benjamin Bercovitz, and Hector Garcia-Molina. Recsplorer: recommendation algorithms based on precedence mining. In *SIGMOD '10: Proceedings of the 2010 international conference on Management of data*, pages 87–98, New York, NY, USA, 2010. ACM.

[27] M. Pazzani and D. Billsus. Learning and revising user profiles: The identification of interesting web sites. *Machine Learning*, 27:313–331, 1997.

[28] P. Resnick, N. Iakovou, M. Sushak, P. Bergstrom, and J. Riedl. Grouplens: An open architecture for collaborative filtering of netnews. In *CSCW*, 1994.

[29] B. Ribeiro-Neto, M. Cristo, P. B. Golgher, and E. S. de Moura. Impedance coupling in content-targeted advertising. In *In SIGIR Š05: Proc. of the 28th annual intl. ACM SIGIR conf.*, page 496Ű503, 2005.

[30] Senjuti Basu Roy, Sihem Amer-Yahia, Ashish Chawla, Gautam Das, and Cong Yu. Constructing and exploring composite items. In *SIGMOD Conference*, pages 843–854, 2010.

[31] Yahoo SmartAds. *http://advertising.yahoo.com/media-kit/smart-ads.html*, 2009.

[32] Jaime Teevan, Susan T. Dumais, and Daniel J. Liebling. To personalize or not to personalize: modeling queries with variation in user intent. In *SIGIR*, pages 163–170, 2008.

[33] Philip Zimmermann. *The Official PGP User's Guide*. MIT Press, 1995.