Building Consumer Trust with Accurate Product Recommendations

A White Paper on LikeMinds WebSell 2.1

Dan R. Greening, Ph.D.
Chief Technical Officer, LikeMinds, Inc.
dan@greening.org
(415) 284-6966

LMWSWP-210-102297

Introduction

LikeMinds WebSell is a web-based system that accumulates a database of consumer product preferences, then uses those preferences to make highly specific customer recommendations for products such as movies, music, articles, books, food, etc.

By finding other people with similar tastes, the system can recommend the most appealing new products to a user with surprising accuracy. Users can search for a specific product, then obtain a predicted rating. WebSell can find 'products for two or more' which satisfy the tastes of multiple users. And, it can find 'worst products,' which in the case of movies, travel, or CDs can be a hilarious diversion.

Users of the LikeMinds Preference Server typically find it addictive and fun. Incorporating the LikeMinds Preference Server can substantially increase the traffic at a web site, and increase customer loyalty to the site by making trustworthy recommendations. Typical Preference Server users average about 50 page views per visit, rate 80 products per visit, receive an average of 40 product recommendations per visit, and exhibit a multi-day repeat rate of greater than 90%.

Using LikeMinds WebSell on retail sales, rental or promotional web sites can increase customer satisfaction substantially. WebSell usually makes much better product recommendations than a close friend would make. Customers learn to trust the system, and by extension learn to trust the web site brand. This strong level of trust brings customers back again and again.

LikeMinds has invested substantial resources to make LikeMinds WebSell extremely accurate. Patents 4,870,579 and 4,996,642, issued in 1989 and 1991 -- before web activity became widespread -- cover the algorithms used in Web Sell and other LikeMinds products. These patents cover the most accurate forms of 'collaborative filtering,' essentially all algorithms which compare the ratings of two or more users and assign a weight based on similarity. We continue to refine the algorithms through ongoing development and testing.

Movie Critic Demonstration Site

LikeMinds has built a WebSell demonstration web site called "Movie Critic." This system is available at http://www.moviecritic.com/. Movie Critic asks users to rate 12 movies, then recommends additional movies a user will like. The more movies a user rates, the better the recommendations as the system refines its understanding of a user's tastes and preferences.

Movie Critic can limit recommendations to specific product categories called 'genres'. Users can obtain recommendations for movies available in video rental stores, or for movies showing in theaters. For instance a WebSell application built for a video web site with retail storefronts could use this feature to limit recommendations by subject, year, or whether a particular movie was in stock.

Another feature of WebSell that is used in Movie Critic is the ability to create 'composite users' from two or more people. This allows Movie Critic to make recommendations that will satisfy everyone in a family or a couple. Movie Critic demonstrates this by providing 'Movies for Two' recommendations -- movies that both you and a friend will

like. You choose another Movie Critic member by typing their user-id. Movie Critic then intersects your rating vector with your friend's to form a composite user, and computes recommendations from that.

Internally, Movie Critic first generates predicted ratings, then provides recommendations. Applications built with WebSell can use this information directly. For example, Movie Critic lets you skip the recommendation step to obtain your predicted rating of any movie. If your friends want you to go to a movie, you can search Movie Critic for the movie and get your predicted rating.

Using predicted ratings, Movie Critic can even recommend 'Worst Bets,' the movies you will like the least. Although this does not drive most viewing decisions, it can be hilarious. People who might watch *Mystery Science Theater 3000*, a popular television program which spoofs bad science fiction movies, appreciate this feature and do use it to suggest rental movies.

Movie Critic computes its confidence in each predicted rating. The confidence level combines the strength of the recommendation (how similar are you and your mentors) and the dissent among your mentors (how controversial is a product among your peer group). Movie Critic provides this information in the form of a bulls-eye. When a confidence arrow hits the center of the bulls-eye, Movie Critic is sure of its prediction. When it misses the center, the Movie Critic is less confident. 'Not confident' predictions also indicate movie ratings that will help narrow down your preferences.

Users of Movie Critic typically find it addictive and fun. User surveys from LikeMinds Movie Critic site prove that incorporating WebSell into your web site can substantially grow traffic, and increase customer loyalty by making trustworthy recommendations. For example, typical Movie Critic users rate 80 products, average 50 page views while receiving 40 product recommendations per visit, and exhibit a multi-day repeat rate of greater than 75%.

The Principles behind WebSell

WebSell has three main functions: First, find out more about a user's product preferences. Second, identify products the user is likely to want. Third, make product recommendations to the user.

People Use Objective and Subjective Factors

A product can be anything on which people express differing opinions. Products include:

- retail items, such as music CDs, video tapes, toys, books, or automobiles,
- experiences, such as travel locations, concerts, or sports events,
- consumables, such as articles, music or video clips,
- people, as might be expressed in a dating service or teleconference,
- job listings, and
- advertisements (both banner ads and classifieds).

We make decisions on products using objective and subjective factors. Objective factors depend solely on the product: Who is the author of a book? What is the native language of a travel destination? How is this music clip classified? How tall is a potential date? How much does a job pay?

Subjective factors depend on us: Do we like the prose style of a book? Will I have fun in a travel destination? Is the music good? Is a potential date cute? Will I like the boss?

Typical classification systems can help us objectively assess products, but to subjectively assess products we rely on personal relationships: Did a salesperson who understands what I'm looking for recommend the book? Did a person like me recommend a vacation spot? Did a reviewer I respect enjoy the musician? Do I like the employer's product advertisements?

Product Preferences Drive Marketing

WebSell improves subjective decisions by using a database of user preferences to inform product recommendations.

Product preferences are the most accurate way to identify what people like. This is an obvious tautology, but consider that vast industries have been formed around surrogates for product preferences.

Demographic analysis is one surrogate. Demographic analysis surveys individuals to get product preferences and 'objective' factors, such as age, income, home address, and marital status. Then it classifies the audience by objective factors to infer product preferences.

For example, in creating a campaign for a new Madonna movie, we look at customer demographics -- 19 to 24 year old unmarried women pay to attend Madonna movies. And we look at media demographics -- the *Beverly Hills 90210* television show has a large audience of 19 to 24 year old unmarried women. The combination tells us we should advertise our new Madonna movie on *Beverly Hills 90210*.

Psychographic analysis is another surrogate. Psychographic analysis surveys individuals to get product preferences and to get answers to psychological questions. The psychological questions classify the individual and help determine appealing products and advertisements.

For example, a 'gatekeeper mom' answers psychological questions in ways that indicate she tries to control her child's environment. Gatekeeper moms are more likely to respond to advertisements that show a loving mother giving a child an apple.

These generalizations provide approximations to true consumer preference groups. To the extent that an approximation fails to represent true individual preferences, we waste marketing dollars targeting people who won't buy the product.

Trust Builds Strong Brands

WebSell leverages a simple principle: Consumer preferences are not random -- people who express the same taste in products can recommend products to each other. People use this principle to make decisions without using computers. When we find someone else who likes the same music, sports or cars, we talk about favorite products to help inform our own future purchases.

Brand trust helps improve decision-making efficiency, in a similar way. We learn that a brand is associated with products we like, and we then begin to trust that brand.

How WebSell Works

WebSell is based on the LikeMinds Preference Server module of the LikeMinds product architecture. Preference Server creates a preference database to make product recommendations better than friends or salespeople can. Here's a simplified description of how it works:

You (a consumer) express your taste in a set of products. You can do this implicitly by viewing web pages (we count clicks or clock viewing time) or explicitly by selecting and rating products. In either case, we call the result a 'rating vector.' The Preference Server assembles the rating vectors of thousands or millions of other people in its database.

The Preference Server then identifies those people most like you. We call people like you 'mentors.' The Preference Server assigns a numeric weight to each mentor based on the similarity of his or her rating vector to yours. The more the mentor's rating values resemble yours and the more products both of you have rated, the greater the weight.

The Preference Server assembles a set of recommendations by finding the products each mentor recommends and creating a 'prediction vector' containing the predicted rating of each product. With each predicted rating, it also stores a numeric 'confidence' for the rating.

The Preference Server recommends products to you by sorting the list of products, then presenting those products you are most likely to want.

Preference Server is Accurate

The general algorithm used in the Preference Server is detailed in U.S. Patents 4,996,642 and 4,870,579.

The basics follow: For any user selected from a group, the system recommends items, such as movies, sampled by one or more users in the group, but not sampled by the selected user.

The system first identifies a user. Movie Critic does this when the user logs into the web site.

The system records a scalar 'rating' for each item sampled by the selected user to represent the user's reaction to that item. Movie Critic does this whenever a user rates a movie. The set of ratings for a user is called the 'rating vector.'

The system pairs the selected user with a random collection of others who have sampled some items that the selected user also sampled. The system computes a scalar 'agreement strength' for each user pair, using the difference in ratings for items sampled by both, and the number of items sampled by both.

The system designates some of the other users as recommending users or 'mentors.' Mentors are users who have stronger agreement, recommend more items, or improve the item coverage of the mentor set.

Preference Server performs these computations with random user pairs in a background process called the 'background recommender,' and with cached users in the 'immediate recommender.'

When a user requests a recommendation or a predicted rating, the system fetches mentors and their rating vectors. The system then assembles a 'prediction vector' which for each item contains the predicted rating of the item, the 'confidence' of the prediction, and the controversy associated with the prediction.

The system then identifies items to recommend to the user. Recommendations can be determined by a combination of the best predicted rating, best confidence, lowest controversy, and availability of the item in the inventory.

Example

To illustrate the calculations, we will walk through a simple movie example. The functions described in this section are not exactly those used in WebSell or Movie Critic, but they are close enough that the algorithm is clear.

Figure 1 show a set of example movie ratings entered by users Smith, Jones, and Wesson. In this example, ratings vary from 13 (best) to 1 (worst). All three users have rated *Star Wars* and *The Untouchables*. Smith has not seen *Beverly Hills Cop*, and Jones has seen neither *Fletch* nor *Caddyshack*.

To compute the agreement of Smith and Jones, we first find the differences in common ratings. Smith and Jones have sampled two items in common -- *Star Wars* and *The Untouchables* -- having rating differences of 3 and 1, respectively.

	Star Wars	The Untouchables	Beverly Hills Cop	Fletch	Caddyshack
Smith	8	10		10	7
Jones	11	9	10		
Wesson	10	4	10	9	11

Figure 1: *Example Ratings*

Rating Difference	Closeness	Rating Difference	Closeness
0	10	7	0
1	9	8	-1
2	6	9	-6
3	4	10	-8
4	2	11	-10
5	1	12	-10
6	0		

Table 1: *Example Closeness Function*

To obtain the agreement strength, we first compute a closeness function of each pair of ratings. Table 1 shows an example closeness function. If the two ratings are exactly the same, the closeness function returns 10. If the two ratings differ by 10, the closeness function returns -8, etc. For our example the closeness function values are 4 and 9, respectively.

We compute the agreement scalar for a user pair with the following equation:

$AS = (CVT/n)\log_2 n$ (*equation 1*)

where AS is the agreement scalar, CVT is the sum of the closeness-values, and n is the count of items sampled by both users.

By application of equation 1, CVT is 13 and the agreement scalar for Smith and Jones is 6.5. Similarly, CVT for the pair of Smith and Wesson is 17 and the agreement scalar is 8.50. CVT for Jones and Wesson is 20, and the agreement scalar is 10.6.

The difference in reaction of Smith and Jones to *The Untouchables*, and the small number of items held in common led to the smaller agreement scalar between those users. The greater the number of items that the users have sampled, the more accurate the agreement scalar should be for each of the users with which the selected user is paired.

		Mentors: Smith	Mentors: Jones	Mentors: Wesson
Selected Users	Smith		6.5	8.5
	Jones	6.5		10.6
	Wesson	8.5	10.6	

Figure 2: *Mentor weights*

The mentors are ranked by order of agreement scalar. In one version of the algorithm, predictions from the higher weight mentor supercede predictions from all lower weight mentors. In another version of the algorithm, competing predictions from different mentors are summed proportionally to their relative weight.

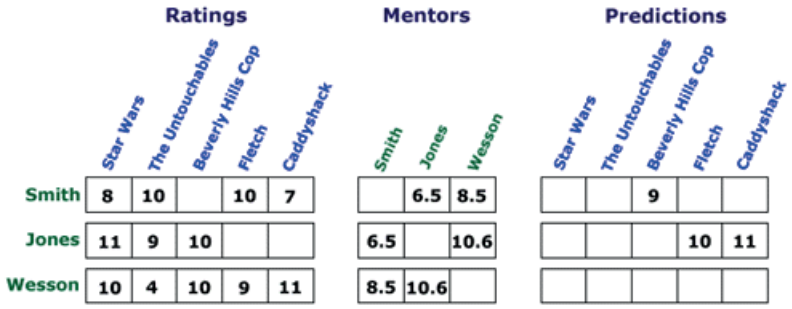

Figure 3: ***Computing Prediction Vectors***

The predictions themselves are not simply copied from the mentor's rating vector. A mentor's ratings are scaled to fit the selected user's range of ratings. This is done to allow mild reviewers to mentor extremists, and visa-versa. Assembling these scaled predictions from all the mentors creates a prediction vector, as illustrated in Figure 3.

With the prediction vector filled in, the system obtains an item recommendation by sorting the items by their predicted rating. The system will recommend *Caddyshack*, with a predicted rating of 11, to Jones. It will recommend *Beverly Hills Cop*, with a predicted rating of 9, to Smith.

Alternatively, the predictions can be sorted in least appealing order, to get a 'Worst Bets' list, as provided in MovieCritic. Or a user can look up the predicted rating for an item.

The terms 'person' and 'user' can be used in the broadest sense to refer to any entity which exhibits a subjective reaction to an item. The Preference Server applies to more than movies, record albums, computer games, television programs, or other consumer items. For example, reactions can be predicted for travel destinations, hotels, or restaurants. Further, predictions among categories can be accomplished, e.g., recommending books based on the ratings of movies.

Through its database interface, WebSell can interact with an inventory management system or maintain an inventory status itself, when inventory is limited. On a video store web site, for example, certain movies may not be carried by that store or, even if the items are stocked they may be unavailable. In these circumstances WebSell can recommend only those items which are available. It can also recommend only specially designated items (such as items on sale, or those within a particular genre).

WebSell Accuracy Metrics

Since WebSell is a key factor in web site brand development, it must provide accurate predictions. Good accuracy metrics help objectively judge prediction algorithms and parameter settings.

Because WebSell predicts ratings, it has a built-in metric for computing its own accuracy. To compute the accuracy of the system, we simply remove a user's product rating, then recompute the prediction vector. We compare the removed rating to the corresponding prediction. The prediction error is the difference between the actual rating and the predicted rating.

Measuring WebSell accuracy involves tradeoffs. You get a better metric by spending more CPU time or using more storage. Rewinding the prediction algorithm farther back expends more resources to create trial predictions, but follows a more representative prediction process.

The fastest and worst accuracy metric removes a product rating from a user's rating vector, but leaves the mentor list and strengths alone. The old mentor list and strengths generate a prediction vector that includes the removed product. The difference between the original rating and the predicted rating is the accuracy. This metric takes the same amount of CPU time required to obtain a typical recommendation vector from the mentor list. It can be performed with no additional memory or database storage.

The medium speed, moderately good accuracy metric removes a product rating from a user's rating vector and *recomputes the mentor strengths*. The old mentor list and new strengths generate a prediction vector that includes the removed product. The difference between the original rating and the predicted rating is the accuracy. This metric takes more CPU time, but better reflects the actual algorithm, so it obtains a better accuracy measure. It can be performed with no additional memory or database storage.

The slowest and best accuracy metric removes a product rating from a user's rating vector, *removes all the old mentors*, *finds new mentors* and *computes their weights*. The new mentor list and new strengths generate a recommendation vector that includes the removed product. The difference between the original rating and the predicted rating is the accuracy. This metric consumes a substantial amount of CPU time and storage, but produces an exact measure of the accuracy.

The fast metrics are valuable for quickly comparing the outcomes of different parameter values. However, the fast metrics suffer from 'self-fulfilling prophecy' -- mentors were selected using the rating that was removed, so they are bound to better predict that rating. The slow metric is valuable for comparing mentor selection algorithms and parameters.

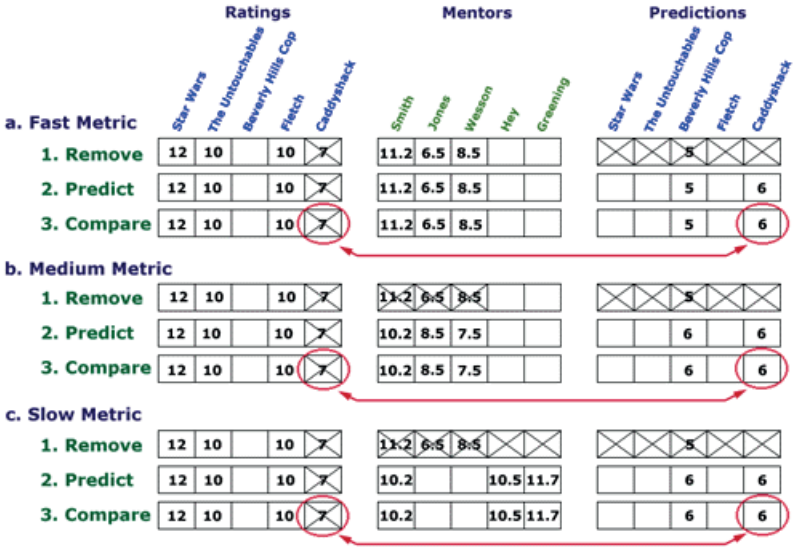

Figure 4: *Accuracy Metrics*

Figure 4 shows how these metrics work. In Figure 4a, we temporarily delete a users' rating of *Caddyshack*, then recompute. In Figure 4b, we temporarily delete and recompute the mentor weights. In Figure 4c, we take the expensive step of finding replacement mentors. In each case, we compare the original rating with the predicted rating to find the error.

WebSell is Broadly Applicable

Cold Start

'Cold start' refers to a situation where no preference information is available and there are no mentors to compare with new users. Until WebSell is seeded with preference information it cannot make recommendations.

In most cases, the absence of seed data is only a temporary problem. Consumers are often eager to give their opinion, even if there is no immediate reward. WebSell promises deferred gratification -- in a day or two consumers will receive highly accurate recommendations.

A WebSell site can improve the database seeding process by leveraging information about the customer. For example, if a site records prior purchases of every user, it can suggest that the user rate products she has purchased in the past. If demographic information is available, the user's demographic can suggest products a user can rate.

However, certain product categories have on-going cold start problems. Some product inventories are so large that it may take a long time to obtain consumer preference information for some products. For example, large bookstore inventories may exceed 1 million books. Only a couple of ratings are required to start recommending an item, so WebSell will start recommending popular books quickly after start-up. However, obscure books may take months to receive two ratings.

Some types of products are transient. For example, articles published weekly or daily are items people want to read, however, the first few people to read an article cannot rely on explicit preference information to recommend articles. Because they are the first to read the article, there are no mentors who can recommend it.

However, WebSell provides a solution.

Marketers typically have access to large databases of information available on their products and their customers. Any characterization that segments the product space can act as seed data for the WebSell. Here are three examples:

- Product-specific information relies on consumer preferences for particular brands, authors, directors, actors, musicians, etc. Consumers who like Madonna's first album have a higher-than-average probability of liking Madonna's last album. Other brands -- such as manufacturer, publisher, director, actor, and musician -- help roughly segment products by consumer preference. Even attributes we don't call 'brands,' such as production year, help segment products. For example, some of us like '60s' music.

- Preference surrogate information relies on traditional consumer analysis to help segment products. For example, demographic and psychographic segmentation can identify people who have a greater probability of liking a product.
- Consumer-specific information relies on consumer behavior to help segment products. Many businesses consider it an important brand strategy to track this information to drive inventory decisions, so the information is already there. For example, a merchant may track the books purchased by each consumer. The consumer probably liked the books she purchased, though this is not guaranteed because the book may be a gift, or she may decide it's a dud after reading it. Nevertheless, purchase behavior provides a good means to segment products by preference when no other consumer-specific information is available.

It turns out that the regular WebSell algorithm can be used directly to incorporate this information.

Pick any attribute which drives consumption behavior, such as "author=Oliver Sacks". Construct a 'virtual user' who loves all products satisfying the attribute, but expresses no opinion about others, and put it in the database. This virtual user then acts like a simple salesperson -- if a consumer says she like an Oliver Sacks book, the salesperson would likely recommend another Oliver Sacks book.

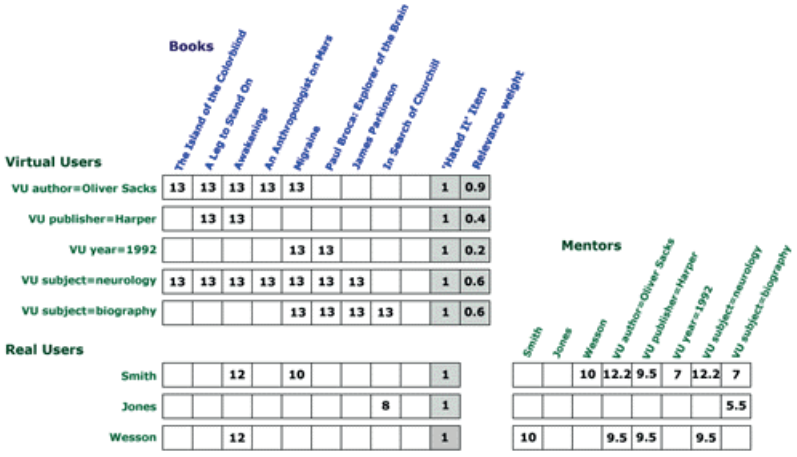

Figure 5: ***Generating Recommendations from Objective Factors***

Figure 5 shows tables from a book recommendation system. Virtual users have been entered for the author Oliver Sacks, publisher Harpers, publication date 1992, and subjects neurology and biography. Real users Smith and Jones have both sampled the book *Awakenings*.

The mentors list indicates our confidence in a recommending user. Wesson's strongest mentor is Smith. Books recommended to Wesson by Smith will have the strongest confidence. Wesson's most confident predicted rating is 10 for *Migraine*.

The strongest mentors for Smith are virtual users *author=Oliver Sacks* and *subject=neurology*. Smith has more in common with a hypothetical person who loves all Oliver Sacks books and with another hypothetical person who loves all neurology books, than with any real person in the database. WebSell will recommend other Oliver Sacks books and other neurology books to Smith.

WebSell provides additional features to help tune 'cold start' ratings. For example, an installation can assign a factor to each virtual user. Suppose publication *year*=1992 had *n*=100 books in the inventory. To avoid assigning a high-confidence to publication year, a reasonable factor might be 1/*n*. Then if someone loved exactly one book published in 1992, recommendations deriving just from this publication date would have confidence 10/100 = 1/10, very low confidence.

An installation can skew recommendations toward stronger confidence or higher ratings. Movie Critic, for example, uses a combination: it will not recommend movies with low confidence or low predicted ratings.

WebSell is Efficient

WebSell is designed to perform well in applications with millions of items and millions of users, with thousands of simultaneous logins.

WebSell includes a cache of recently used mentors. Since most configurations prefer mentors that rate large numbers of items, and those representing popular opinion, those users are more likely to be in the cache. Because of normal statistical distributions, 'average people' get recommendations fast from WebSell's cache.

The WebSell design allows for efficient parallel processing. The most important factor in creating efficient parallel processing algorithms is limiting inter-processor communication. Inter-processor communication appears explicitly through message-passing functions, or implicitly through shared data structures and synchronization functions.

The inter-processor communication required to identify and weight mentors (in background processes) is limited to the synchronization necessary to prevent two background processes from simultaneously updating the same user's mentor list. This is handled by simply assigning disjoint lists of users to different processes.

The inter-processor communication required to compute predicted ratings for a user who has logged in is limited to that required to keep the user from logging in again to a different processor. Mentor ratings used in predicting ratings can be *stale* without damaging the accuracy of the algorithm. Therefore, the algorithm requires no synchronization to keep mentor ratings up to date. To improve the efficiency of the algorithm, cached mentors have a time-stamp, and the cache purges them after an expiration time.

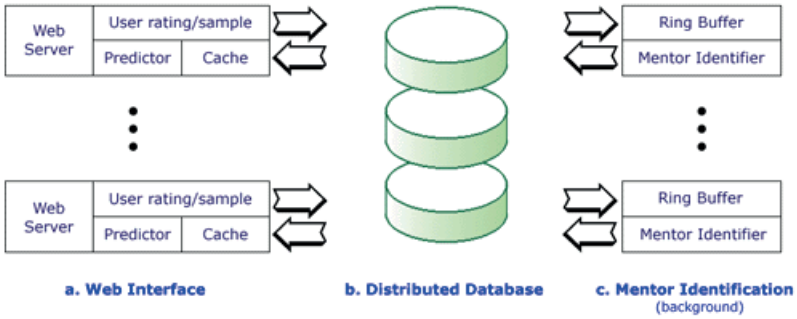

Figure 6: *Parallelism in Preference Server 2.1*

Figure 6 shows how a site can distribute WebSell tasks over multiple processors. Figure 6b shows that as the simultaneous users increase, the site can add new web servers with associated web processing. Efficiency requires that once a user logs into WebSell, foreground processing for him or her must remain on the same processor. This is handled by naming a specific web server following login. For example, the login process might begin using a web request distributor, say www.bigcorp.com, but following successful login would refer to the web server holding the user's data, say www7.bigcorp.com.

Figure 6b shows that using WebSell's ODBC or Oracle interface, the WebSell can exploit a distributed database. A site should use a distributed database when the number of simultaneous users increases.

Figure 6c shows that increasing the number of background processes increases the rate at which mentors are identified and introduced into the database. A site should add background processes when the number of registered users increases, or if the site needs more accurate predictions to be established faster.

Resource Requirements

Several factors affect resource requirements of WebSell:

u = total number of users,

p = total number of products,

r = average number of rated products per user,

m = maximum number of mentors per user,

s = maximum simultaneous users.

As u increases, disk storage increases by the size of the user database record.

The background processing time follows the relation $O(t) = u$. This processing time can be split among several processors as described previously. If the algorithm required comparing each user to every other user, the background processing time would increase by u^2, a large number for many installations. Fortunately, WebSell need only compare a fixed number of old users to each new user.

Performance improvements can be obtained by directing the algorithm to spend more background processing time on recently changed users. However, note that some background processing must be devoted to improving the mentor lists for older users records, otherwise these users will not gain the benefits of new user ratings.

Database storage required follows this relation: $O(d) = u + p + ur + um$.

Cache memory storage required follows this relation: $O(c) = rm$. This storage is required in each processor.

Predicted rating computation time, follows this relation: $O(t) = rms$. This processing time can be split among up to s processors.

WebSell Integrates with Existing Systems

LikeMinds products are designed to interface with a broad range of web servers, database servers, operating systems, and programming languages. At the same time, we focus on efficiency -- we provide fast proprietary interfaces, as well as generic interfaces.

Web Server Interface

The WebSell runs on any web server that supports CGI (Common Gateway Interface). Due to process and socket overhead, CGI can be slow. To eliminate these bottlenecks, the WebSell comes with a "Proxy ODBC Daemon" which holds database connections and processes open as transient CGI programs start and stop.

The WebSell also supports a non-proprietary interface called 'FastCGI,' which is often faster than proprietary web server interfaces. Apache, OpenMarket, and other servers provide support for FastCGI. Source code is freely available from LikeMinds or the FastCGI Consortium.

WebSell also supports several proprietary interfaces to popular web servers, such as Netscape Enterprise Server and Microsoft Site Server.

Finally, programmers can access the Preference Server C++ API directly for highly customized applications. The C++ interface rests on top of the Preference Server Library, which contains the Preference Server correlation engine. Using this interface, customers can create a variety of applications, including non-web based applications (such as consumer kiosks), or applications that interact with proprietary networking protocols.

Web Page Interface

The web interface consists of HTML templates that organize a user's online interactions with the Preference Server. These templates can be customized to create a distinctive look and feel for the web site. Server-side Java and JavaScript APIs can also be used to create a custom look-and-feel.

Database Interface

Many sites already have extensive databases with information on customers and products. They don't want to change database servers or data formats. LikeMinds WebSell interacts with existing database servers and schemas, making it easy to add to an existing site.

LikeMinds WebSell can be configured to work with any existing database format through ODBC, the Open Database Connectivity standard. ODBC lets programs access an SQL relational database regardless of the database vendor's API. LikeMinds WebSell supports all popular database systems, including Informix, Oracle, Sybase, Microsoft, etc. For operating systems where no ODBC client-side API is available or the ODBC driver is inadequate, WebSell can directly call the database API. A direct interface currently exists for Oracle databases, as well as others.

Conclusion

WebSell converts web surfers into loyal customers because it delivers trustworthy recommendations for products. Consumers are frequently surprised by product recommendations they might not have expected, but after a successful product experience, their trust in a web site increases.

LikeMinds has surveyed the Movie Critic population to determine the utility and desirability of WebSell-based applications. Respondents identified several reasons for using Movie Critic. The Movie Critic saved them time and money in choosing movies to see. Movie Critic helped them see more movies they were confident they would like. Movie Critic helped them see fewer 'duds.' A large majority said it was 'just plain fun' to interact with Movie Critic.

When asked to rate their overall satisfaction with Movie Critic, almost all responded with 'love it' or 'really good'. This overwhelming user satisfaction with a WebSell-based application is a testament to the accuracy of the LikeMinds Preference Server technology.

WebSell allows for mixing subjective impressions -- such as user ratings, click-throughs, and usage time -- with objective attributes, such as author, subject, keywords, and demographic appeal. By making recommendations based on both subjective and objective criteria, merchants can ensure trustworthy recommendations even in 'cold start' situations.

The Preference Server module upon which WebSell is based is an inherently scaleable algorithm. It can handle large databases of users and products with linear resource increases. It is designed for multiprocessing, and can exploit data reference locality through a specialized mentor cache.

Administrators can customize WebSell to create a brand-consistent look-and-feel through HTML, CGI, Java, or JavaScript APIs. They can connect existing product and user databases with WebSell to add value to an existing customer base.

WebSell has been designed and built as a marketing and sales tool for commercial web sites. Our attention to trustworthiness, broad applicability, performance, and customization reflects our focus on commercial web site needs.
