In this paper, we propose an XML-based recommender system, called SPGProfile. It is a type of collaborative information filtering system. SPGProfile uses ontology-driven social networks, where nodes represent social groups. A social group is an entity that defines a group based on demographic, ethnic, cultural, religious, age, or other characteristics. In the SPGProfile framework, query results are filtered and ranked based on the preferences of the social groups to which the user belongs. If the user belongs to social group $G_x$, results will be filtered based on the preferences of $G_x$ and the preferences of each ancestor social group of $G_x$ in the social network. SPGProfile can be used for various practical applications, such as Internet or other businesses that market preference-driven products. In the ontology, the preferences of a social group are identified either: (1) by the preferences of its member users, or (2) from published studies about the social group. We describe and experimentally compare these two approaches. We also experimentally evaluate the search effectiveness and efficiency of SPGProfile and compare it to two existing search engines.

1. Introduction

Traditional search engines typically return identical results for the same query, independent of the user or the context. Conventional quantitative scoring functions may not adequately reflect users’ preferences, since the same document may be queried by users, whose preferences differ. By analyzing search behavior, it is possible to see that many users are not able to accurately express their needs in exact query terms [32]. In contrast to conventional search engines, a personalized search engine [22, 6, 43] would return different results for the same query, depending on the user and the context. Profiles can modify the representation of the user needs before the retrieval takes place. Most personalized systems lean towards being Information Filtering (IF) systems more than being general Information Retrieval (IR) systems [36].

Most existing personalized search systems do not consider group profiling. Group profiling can be an efficient retrieval mechanism, where a user profile is inferred from the profile of the social groups to which the user belongs. We propose in this paper an XML search system called SPGProfile (Speak Group Profile), which employs the concept of group profiling. SPGProfile simplifies the personalization process by pre-defining various categories of social groups and then identifying their preferences. The framework of SPGProfile categorizes social groups based on demographic, ethnic, cultural, religious, age, or other characteristics. For example, people of ethnic group $E_X$; people who follow religion $R_Y$; and people who live in neighborhood $G_Y$ can all be considered to form various social groups. In social communities, it is commonly accepted that people who are known to share a specific background are likely to have additional connected interests [19]. SPGProfile can be used for various practical applications, such as Internet or other businesses that market preference-driven products. An individual user may belong to more than one social group. Therefore, SPGProfile outputs ranked lists of content items, taking into account not only the initial preferences of the user, but also the preferences of the user’s various social groups. Consider for example a Mexican-American user. The user belongs to social groups Mexicans and Americans: the portion of Mexicans living in the USA. The results of a query submitted by this user will be filtered and ranked based on the union of the interests of social groups “Mexicans” and “Americans”. The social groups to which a user belongs usually have class-subclass relationships. A subclass social group has its own properties while inheriting the properties of its superclass(es). For example, consider a user who belongs to the ethnic group “Berbers”, who lives in the country of “Morocco”, which is part of “North Africa”. We could have the following representation of the hierarchical relationships between the three social groups: North Africans $\rightarrow$ Moroccans $\rightarrow$ Berbers. The Berbers may have their own concerns and preferences, while sharing the concerns and preferences of Moroccans, and more general concerns and preferences of North Africans. Thus, the user’s query will be filtered and ranked based on the preferences of “Berbers”, “Moroccans”, and “North Africans”.

SPGProfile: Speak Group Profile

Kamal Taha and Ramez Elmasri
701 S. Nedderman Drive. Department of Computer Science and Engineering, the University of Texas, Arlington, Texas 76019. E-mail: {kamal.taha, elmasri}@uta.edu
In the framework of SPGProfile, the preferences of a social group could be identified from either: (1) the preferences of its member users, or (2) from published studies about the social group (the availability of such data has had a significant boost with the emergence of the World Wide Web). We describe and experimentally compare the two approaches in this study. SPGProfile helps searchers find what they need faster by trying to put the most relevant results at the top of the result list, a process known as relevance ranking. To the best of our knowledge, this is the first work advocating filtering and ranking results based on the preferences of social groups defined based on characteristics such as ethnicity, culture, religion, or the like. We make the following contributions in the paper:

- We experimentally evaluate the search effectiveness and efficiency of SPGProfile and compare it to two existing search engines. A demo of the SPGProfile system, running on 1000 MB grocery data and 1000 MB car data, is available at: http://dbse1.uta.edu/~kamal/?action=home.
- We propose ontology-driven social networks for modeling social groups.
- We propose rule-chaining (recursive querying) techniques for filtering query results based on the preferences of users’ social groups.

2. Concepts used in the paper

In this section, we define key notations and basic concepts used in this paper. We use the term “domain” throughout the paper to mean an area of activity, belief, culture, ethnicity, demography, pursuit, or the like. A Single-Domain Group (SDG) is a group of people sharing common domain interests. For example, people of ethnic group $E_X$ represents a SDG. We now formalize the concept of SDG.

**Definition 1 - Single-Domain Group (SDG):**

A SDG is an aggregation $G$ of individual users, where for each $x, y \in G$ ($x \neq y$): $x$ and $y$ share a common and distinctive culture, ethnicity, religion, demography, language, or the like. That is, $x$ and $y$ share the same interests of only one domain group.

We use the term “system administrator” throughout the paper to mean a person employed to maintain and operate a system. The system administrator predefines SDGs, which are usually defined in publications such as: (1) published government’s census and statistical studies (e.g., [42, 30]), and (2) published studies conducted by specialized centers belonging to universities and organizations (e.g., [10, 27, 28, 25]).

The smaller a social group is, the more granular and specific its interests are. Therefore, we introduce another class of social groups called a Multi-Domain Group (MDG), whose size is usually smaller than a SDG. A MDG is formed from an aggregation of people sharing common multi-domain interests. Thus, a MDG is formed from the intersection of two or more SDGs. For example, the portion of ethnic group $E_X$ who follow religion $R_Y$ and live in neighborhood $N_I$ forms a MDG: the intersection of $E_X \cap R_Y \cap N_I$. The interests of a MDG are the union of the interests of the SDGs forming it. Thus, the interests of a MDG are more specific than the interests of each of the SDGs forming it. To fine-grain a user’s query results, SPGProfile outputs a filtered and ranked list of items taking into account the preferences of the user’s MDG. We now formalize the concept of MDG.

**Definition 2 - Multi-Domain Group (MDG):**

Let $S$ be the set of all SDGs that exist. A MDG is an aggregation $G$ of individual users, where: $\forall x \in G$: $x$ shares the same interests of $\exists s \in S : |s| \geq 2$. That is, $G$ is formed from the intersection $\bigcap_{SDG_i \in S : x \subseteq S} SDG_i$.

The interests of $G$ are the union $\bigcup_{SDG_i \in S : x \subseteq S} \text{interest}(SDG_i)$.

SDGs forming a MDG usually have class-subclass relationships, where a subclass has its own properties while inheriting the properties of its superclass(s).
The preferences (interests) of a SDG \( G \) are stored in SPGProfile’s database in the form of a trigger rule, called \( \text{TrigRule}(G) \). In response to a user’s query, SPGProfile triggers the trigger rules of the SDGs forming the user’s MDG. The trigger rules filter the XML tuples, retaining only those satisfying the preferences of the SDGs. The construct of a trigger rule is formed from the “WHERE” clause of XQuery [7]. That is, a trigger rule contains predicate Boolean conditions and these conditions are the preferences of a SDG. Fig. 1 shows a form of a trigger rule. The symbol \( \Delta \) denotes either an XQuery’s child operator ‘/’ or a descendant operator ‘//’. The symbol \( \circ \) denotes an XQuery comparison operator. The letter \( P \) denotes a preference of a SDG corresponding to the value contained in an element labeled \( L \) in the XML document.

\[ \text{s} \Delta L \quad \circ \quad P \]

and/or/not  

\[ \text{s} \Delta L \quad \circ \quad P \]

FIG. 1: Form of a trigger rule

3. Modeling SDGs and MDGs

3.1 Modeling SDGs

We model the ontological relationships between SDGs using entity-based domain ontology modeling, which we call group profile ontology. In this modeling technique, the relationships between SDGs of the same domain are represented by their ontological relationships. For example, in religion-based domain, the ontological relationships between some of the branches of Buddhism are represented as follows: “Buddhists” \( \rightarrow \) “Mahayanists” \( \rightarrow \) “Zens”: “Zens” SDG is a subclass of “Mahayanists” SDG, which is a subclass of “Buddhists” SDG. SDGs of different domains could be related (linked) by an interoperable SDG\(^1\): for example, SDGs of domains \( D_i \) and \( D_j \) could be linked by a SDG of domain \( D_x \) (\( D_x \neq D_i, D_j \)).

The creation of group profile ontology is done semi-automatically. SPGProfile prompts the system administrator with two text fields, one representing a class SDG and the other its immediate-subclass SDG. After the system is informed of each two SDGs having class-immediate subclass relationship, it creates a group profile ontology in the form of an OWL ontology [4]. The OWL file defines the relations between SDGs as ontological classes. Thereafter, SPGProfile converts the OWL file into ontology-driven graphical representation, called Single Domain Graph (SDGraph). In a SDGraph, each SDG in the OWL file is represented by a vertex, and each class-subClassOf relation is represented by a directed edge. That is, for each two vertices \( u \) and \( v \), there is an edge \((u, v)\) in the SDGraph if \( v \) is a subClassOf of \( u \) in the OWL file. We now formalize the SDGraph concept.

**Definition 3 - Single Domain Graph (SDGraph):**

A SDGraph is a pair of sets \((V, E)\), where \( V \) is a finite set of vertices representing SDGs and \( E \), the set of edges, is a binary relation on \( V \), so that \( E \subseteq V \times V \). Let \( \pi[u] \) denote the set of vertices representing SDGs that are immediate subclasses of the SDG represented by vertex \( u \in V \). \( E = \{\text{edge } (u, v): u \in V \text{ and } v \in \pi[u]\} \).

SPGProfile internally represents a SDGraph \((V, E)\), as a collection of adjacency lists. An adjacency-list representation of SDGraph is an array \( Adj \), which consists of \(|V|\) lists, one for each vertex in \( V \). For each vertex \( u \in V \), the adjacency list \( Adj[u] \) contains all the vertices \( v \) such that there is an edge \((u, v)\in E\). Alternatively, it may contain pointers to these vertices. The vertices in an adjacency list are typically stored in an arbitrary order. SPGProfile constructs an adjacency list from an input OWL file using SPARQL [37]. SPARQL is a query language for pattern matching against RDF graphs. Variables in the RDF terms will be substituted in the CONSTRUCT part to populate a 2-dimensional array storing the adjacency lists.

\(^1\) An interoperable SDG is a SDG that has ontological relationships with the SDGs, which it links.
Example 1: We use as a running example throughout the paper USA-based SDGs of four domains. These SDGs are: (1) ethnic groups \(E_x\) and \(E_y\), (2) religious groups \(R_x\) and \(R_y\), (3) national origin group \(O_X\), (4) region-based groups \(N_X\) and \(N_Y\) (the people living in neighborhoods \(N_X\) and \(N_Y\) respectively), and (5) region-based groups MPLS and MN (the people living in the city of Minneapolis and in the state of Minnesota respectively). \(N_X\) and \(N_Y\) are neighborhoods in MPLS. Ethnic group \(E_Y\) lives in \(N_X\), and follows religion \(R_Y\). Part of \(E_X\) follows religion \(R_X\), and the other follows \(R_Y\). Fig. A in Appendix A shows the group profile ontology of the described SDGs, in the form of OWL ontology. Fig. 2 shows a SDGraph constructed from the OWL ontology.

3.2 Modeling MDGs

A MDG is denoted by the set of the SDGs forming it. SPGProfile analyzes the structure of a SDGraph to identify all possible MDGs that exist because of the interrelations between SDGs. MDGs are constructed as follows: (1) the intersection of each two SDGs connected by an edge and belonging to different domains forms a MDG, and (2) all unique combinations of the MDGs resulted from construction 1 are enumerated; if a combination does not contain two or more SDGs with the same domain, the intersection of its SDGs forms a MDG.

A user may belong to more than one MDG. Therefore, the system needs a mechanism for modeling MDGs to enable it to determine the user’s smallest MDG. We model MDGs using a graphical representation of social links called Multi Domain Graph (MDGraph). In a MDGraph, MDGs are represented by vertices and the ontological relationships between them are represented as edges. Each hierarchical level of the graph contains MDGs formed from the same number of SDGs. A MDG in level \(i\) is formed from \(i + 1\) SDGs. A MDG \(G_x\), in level \(j\) and a MDG \(G_y\), in level \(j + 1\) are connected by an edge, if \(G_x\) and \(G_y\) contain at least one common SDG. We now formalize the MDGraph concept.

Definition 4, Multi Domain Graph (MDGraph):

A MDGraph is an ontology-driven graphical representation of social links. It is a pair of sets \((V, E)\), where \(V\) is a finite set of vertices representing MDGs and \(E\), the set of edges, is a binary relation on \(V\), so that \(E \subseteq V \times V\). \(E = \{\text{edge}(v_i, v_j) : v_i, v_j \in V \text{ and } SDG G_x \in v_i, v_j \text{ and } v_i \text{ in level } n \text{ and } v_j \text{ in level } n + 1\}\). That is, there is an edge \((v_i, v_j) \in E\) if: (1) there exists at least one common SDG in \(v_i\) and \(v_j\), and (2) \(v_i\) and \(v_j\) are located in two adjacent hierarchical levels.

Fig. 3 shows a MDGraph constructed from the SDGraph in Fig. 2. We built an Algorithm called ConstructMDGraph, which constructs a MDGraph. Its input is a SDGraph represented by an adjacency-list. The Algorithm is shown in Fig. B in Appendix B.

SPGProfile determines the users’ MDGs by traversing the paths of the MDGraph starting from the vertices representing the user’s SDGs. The user’s smallest MDG is located in the intersection of the longest paths originated from user’s SDGs vertices. If the paths originated from \(n\) SDGs vertices, the user’s smallest MDG is usually formed from \(m\) SDGs, where \(m > n\) due to the interrelations between SDGs. We now present example 2 to illustrate how a user’s smallest MDG is located.

Example 2: Consider a user whose national origin is \(O_X\) and lives in neighborhood \(N_X\). Using the MDGraph in Fig. 3, SPGProfile can determine that the user’s smallest MDG is \(\{O_X, R_Y, E_Y, N_X\}\) (see level 3). The MDG is located in the intersection of the longest paths originated from the root vertices \(N_X\) and \(O_X\) (the paths are marked with red dashed arrows in Fig. 3). As can be seen, the system started the search using only two SDGs and it could locate a MDG formed from four SDGs.

Since the smaller a MDG is, the more granular its interests are, locating the user’s smallest MDG enables the system to return the most relevant results.

The more SDGs a MDG is formed from, the smaller in size it becomes.
FIG. 2: A SDGraph depicting the relationships between the USA-based SDGs used in the running example.

FIG. 3: A MDGraph constructed from the SDGraph in Fig. 2.
4. Determining and Storing the Preferences of SDGs

The preferences of a SDG could be identified from either the preferences of its member users or from published studies about the SDG. We will describe the two approaches in the next two subsections, but we first define key notations.

**Notation 1 – An item feature:**

We use the term “feature” to refer to the name of a data element in an XML document of preference-driven products. A feature reveals a distinctive and essential aspect of an item.

**Notation 2 – A feature characteristic:**

We use the term “characteristic” to refer to a data item (value) describing a particular data element (item feature) in an XML document. For example, “Ford” is a characteristic of item feature “Make” in the cars XML document. That is, a characteristic is a property which characterizes an item feature.

4.1 Identifying the Preferences of a SDG from the Preferences of its Member Users

We model a user’s preferences as a set $D$ of $m$-triples, $D = \{(a_1, v_1, w_1), \ldots, (a_m, v_m, w_m)\}$, where: $a_i$ denotes item feature $i$, $v_i$ a characteristic of item feature $i$, and $w_i$ a weight on characteristic $v_i$. The weight $w_i$ is a value scaled between 0 and 1. A complete set of item features are presented to users to determine the relevance of items. The preferences of a SDG are particular characteristics of item features, which are deemed important to the user members of the SDG. We adopt the following strategies for determining these preferences:

Each feature is assigned a score. The score is based on the difference between the number of times a feature beats other features, and the number of times it loses. Let $a \succ b$ denote: the number of times that feature $a$ is considered a preference by the members of a SDG is greater than that of feature $b$. The score $c(a)$ of feature $a$ is computed as described in Definition 5.

**Definition 5 – A score of a feature:**

Given a dominance relation $\succ$ on a set $F$ of features, the score $c(a)$ of an alternative feature “$a$” equals $|\{b \in F : a \succ b\}| - |\{b \in F : b \succ a\}|$.

The features’ scores in the subset $F' \subset F$ are the maximal if every feature in $F'$ dominates every feature not in $F'$. This concept is formalized in Definition 6.

**Definition 6 – Dominant features:**

The subset $F' \subset F$ of alternative features with maximal scores is given by $\{a \in F : c(a) \geq c(b), \text{for all } b \in F\}$.

For each feature $f \in F'$, the characteristic $v \in f$ is considered a preference for the SDG, if its preference weight is greater than that of each other characteristic $v' \in f$. This concept is formalized in Definition 7.

**Definition 7 – A preference of a SDG:**

Let $w$ denote the average preference weight on a feature’s characteristic. For each feature $f \in F'$, the characteristic $v \in f$ is considered a preference for the SDG if for each other characteristic $v' \in f$, $w(v) \geq w(v')$.

Preference data could be obtained as follows. The system provides users with a graphical user interface (GUI) to reveal their initial preferences and weights on characteristics. The GUI can consist of graphical elements such as: (1) check boxes representing item features, (2) a drop-down menu associated with each check box for selecting a feature’s characteristics, and (3) optional text fields for inputting preference weights on characteristics.
After the preferences of a SDG have been determined, they would be stored in the database in the form of trigger rules (recall Fig. 1). We now present example 3 to illustrate how preferences are stored as trigger rules.

**Example 3:** Consider the XML document fragment in Fig. C1 in Appendix C and the USA-based SDGs of our running example. Consider that after applying the strategies described above we identified the following preferences: (1) characteristic “spicy” of feature “flavor” is a preference of ethnic group $E_y$, and (2) characteristic “no pork-related products” of “ingredients” is a preference of religious group $R_x$ (e.g., the teachings of religion $R_x$ dictate that). These preferences will be stored in the database in the form of trigger rules FoodTrigRule($E_y$) and FoodTrigRule($R_x$), as shown in Figs. 4 and 5.

<table>
<thead>
<tr>
<th>FoodTrigRule($E_y$)</th>
<th>${ \text{flavor = “spicy”} } $</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIG. 4: FoodTrigRule($E_y$)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FoodTrigRule($R_x$)</th>
<th>${ \text{contains (ingredients, “no pork-related products”)} }$</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIG. 5: FoodTrigRule($R_x$)</td>
<td></td>
</tr>
</tbody>
</table>

### 4.2 Identifying the Preferences of a SDG from Published Studies

The preferences of a SDG can be obtained from published studies such as:

1. Published articles and books (e.g. [2, 20, 33, 5, 24, 39]).
2. Published studies conducted by organizations (e.g. [12]), or specialized centers belonging to universities.

Preferences on an item’s features and characteristics obtained from publications are represented as vectors of weights scaled between 0 and 1. A feature (or a characteristic) is rated based on the consensus of publications on its importance to the SDG. Let $f_n$ be a feature and $c_x$ be one of its characteristics. The overall preference weight on characteristic $c_x$ is computed based on: (1) the ratio of the weight of $f_n$ to the weights of the other features, and (2) the ratio of the weight of $c_x$ to the weights of the other characteristics of $f_n$. First, we need to decide on which publications we will be using\(^4\). For maintaining the integrity of relativity, all rival features (or characteristics) should be rated using the *same* publications. Equation 1 computes the normalized preference weight $\bar{w}_{c_x}$ of characteristic $c_x$.

\[
\bar{w}_{c_x} = w_{f_n} \times w_{c_x} \\
\]

\[
w_{f_n} = \frac{p_{f_n}}{\sum_{f_i \in F} p_{f_i}} \quad \text{and} \quad w_{c_x} = \frac{p_{c_x}}{\sum_{c_{i'} \in f_n} p_{c_{i'}}} \\
\]

- $w_{f_n}$ - the weight of feature $f_n$ with respect to other features.
- $w_{c_x}$ - the weight of characteristic $c_x$ with respect to the other characteristics of feature $f_n$.
- $p_{f_n}$ - the percentage of publications recommending feature $f_n$.
- $F$ - the set of all features.
- $p_{c_x}$ - the percentage of publications recommending characteristic $c_x$.
- $p_{c_{i'}}$ - the percentage of publications recommending a characteristic $c_{i'}$ of feature $f_n$.

\(^4\)The more publications used the more accurate results can be obtained. We also need to select ones issued by reputable sources.
The preferences of a SDG are the characteristics whose weights are equal to or greater than $n^p$, where: $n$ is the number of all characteristics and $p$ is a parameter that can be set to a value in the range 0 to 1. These preferences will be stored in the database in the form of trigger rules.

Example 4: Consider the XML document fragment in Fig. C2 in Appendix C. Let us determine the car preferences of the residents of neighborhood $N_x$ (recall Fig. 2). While some of the preferences of $N_x$ are specific to the neighborhood, others are shared with the people living in the state of MN. According to published surveys, 68% of Minnesotans prefer cars with snow-proof features\(^5\) and 61% prefer fuel-efficient cars\(^6\). Another survey shows that 76% of the residents of $N_x$ prefer cost-efficient cars\(^7\). The characteristics of these three features are recommended by publications, as follows:

a) Snow-proof feature: 25 automobile publications (e.g., [8]) recommend the following snow-proof characteristics:
   - 13 of the publications recommend cars with Electronic Stability Control (ESC).
   - 5 of the publications recommend four-wheel drive (4WD) cars.
   - 7 of the publications recommend both ESC and 4WD.
   Thus, 80% of the publications recommend ESC and 48% recommend 4WD.

b) Cost-efficient feature: According to published census surveys (e.g., [30]), the median income of neighborhood $N_x$ is $60000. Seven consumer and economic surveys (e.g., [3]) found that people prefer buying cars that cost equal or less than 30% of their median income. Accordingly, most of the residents of $N_x$ would prefer buying cars that cost $18000 or less. However, other three different surveys found that people prefer buying cars that cost between 30% and 40% of their median income. Accordingly, most of the residents of $N_x$ would prefer buying cars that cost between $18000 and $24000.
   Thus, the residents of $N_x$ would prefer cars that cost $18000 or less according to 70% of the surveys, and would prefer cars that cost between $18000 and $24000 according to 30% of the surveys.

c) Fuel-efficient feature: 30 fuel-economy guides (e.g., [40]) recommend the following characteristics for fuel efficiency:
   - 12 of the guides recommend ethanol flexible-fuel cars.
   - 5 of the guides recommend hybrid-electric cars.
   - 7 of the guides recommend cars whose Mileage per Gallon (MPG) is higher than 34.
   - 6 of the guides recommend both ethanol flexible-fuel cars and cars whose MPG is higher than 34.
   Thus, 60% of the guides recommend ethanol flexible-fuel cars, 43% recommend cars whose MPG is higher than 34, and 17% recommend hybrid-electric cars.

Table 1 shows the preference weights ($w_{cx}$) on the 7 characteristics after applying Equation 1. The characteristics whose weights $\geq n^p$ (by setting $p$ to 0.1), which are considered preferences of neighborhood $N_x$ are: (1) ESC, and (2) price $\leq $16200. These preferences will be stored in the database in the form of trigger rule CarTrigRule($N_x$) as shown in Fig. 6.

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>ESC</th>
<th>4WD</th>
<th>Price $\leq$ $18000</th>
<th>Price range from $18000$ to $24000$</th>
<th>ethanol flexible-fuel</th>
<th>hybrid-electric</th>
<th>MPG &gt; 34</th>
</tr>
</thead>
<tbody>
<tr>
<td>$w_{f_{x}}$</td>
<td>0.33</td>
<td>0.33</td>
<td>0.37</td>
<td>0.37</td>
<td>0.30</td>
<td>0.30</td>
<td>0.30</td>
</tr>
<tr>
<td>$w_{e_{x}}$</td>
<td>0.63</td>
<td>0.38</td>
<td>0.70</td>
<td>0.30</td>
<td>0.50</td>
<td>0.14</td>
<td>0.36</td>
</tr>
<tr>
<td>$w_{c_{x}}$</td>
<td>0.21</td>
<td>0.12</td>
<td>0.26</td>
<td>0.11</td>
<td>0.15</td>
<td>0.04</td>
<td>0.11</td>
</tr>
</tbody>
</table>

---

5 Due to the very snowy winter in MN.
6 Which is due, in part, to the fact that the government of MN offers sales tax break incentive for buying fuel-efficient cars (e.g., [41]).
7 Due to the fact that $N_x$ is a middle-class neighborhood (e.g., [30]).
5. Answering Queries

In response to a user’s query, SPGProfile first constructs an equivalent XQuery query [7]. Then, it identifies the user’s MDG. Finally, it filters and ranks the initial results based on the preferences of the user’s MDG.

5.1 Identifying the User’s MDG

To identify the user’s smallest MDG, SPGProfile needs to identify at least one of the SDGs to which the user belongs. The more SDGs identified, the smaller in size a MDG can be identified by the system. SPGProfile adopts the following approaches for identifying users’ SDGs: (1) it employs lookup databases provided by [21, 29] to implicitly identify all region-based SDGs (by translating users’ IP addresses to US zip codes, neighborhoods, cities, and states), (2) it encourages (but does not require) users to reveal some of their non region-based SDGs, and (3) it identifies some of the users’ non region-based SDGs implicitly, by analyzing the structure of the SDGraph\(^8\). After identifying the SDGs, SPGProfile determines the users’ MDGs using the technique described previously in subsection Modeling MDGs.

5.2 Filtering and Ranking Results

SPGProfile uses Algorithm FilterResults (see Fig. 7-a) to filter results based on the preferences of the user’s MDG. The Algorithm employs recursive querying (rule-chaining mechanism) to sequentially optimize (filter) results. In each optimization sequence, the results are filtered based on the preferences of one of the SDGs forming the user’s MDG. That is, in each optimization sequence the Algorithm triggers trigger rule TrigRule\((G_x)\), where \(G_x\) is one of the SDGs forming the user’s MDG. Recursive querying allows a query to query the results produced by a previous application of itself and allows the computation of transitive closure of an XML document.

The inputs to Algorithm FilterResults are query \(Q\) (which represents the user’s initial preferences) and the user’s MDG \(V_y\). In line 1 of the Algorithm, function GetInitialResults uses an XQuery search engine to return the IDs of the items, which satisfy the conditions of query \(Q\)^9. These IDs will be stored in a set called \(S\)items. Line 3 filters the IDs in set \(S\)items recursively by calling subroutine RefineSelection (see Fig. 7-b). Only the IDs of the items satisfying the preferences of SDG \(G_x \in V_y\) are retained in each recursion. Line 1 of subroutine RefineSelection iterates over the IDs in set \(S\)items. In each iteration, line 2 stores one of the IDs in a variable called \(CurrentID\). Lines 3-6 are XQuery’s FLWOR expressions. The “where” clause searches for items in the XML document, which satisfy the following two conditions: (1) their IDs match the ID in variable \(CurrentID\), and (2) they satisfy the preferences that are triggered by trigger rule TrigRule\((G_x)\).

---

\(\text{CarTrigRule}(N_x) \{ \begin{align*} &$b$/\text{safety-feature} = \text{“ESC”} \quad \text{and} \quad \$b$/\text{Price} \leq 18000 \end{align*} \}

\(\text{FIG. 6: CarTrigRule}(N_x)\)

---

\(\text{FilterResults}(Q, V_y) \{ \begin{align*} &\text{Sitems } \leftarrow \text{GetInitialResults}(Q) \\
&\text{for each SDG } G_x \in V_y \\
&\quad \text{Sitems } \leftarrow \text{RefineSelection}(G_x, \text{Sitems}) \\
\} \)

\(\text{(a)}\)

\(\text{RefineSelection}(G_x, \text{Sitems}) \{ \begin{align*} &\text{for } (i = 1 \rightarrow |\text{Sitems}|) \{ \\
&\quad \text{CurrentID } \leftarrow \text{Sitems}[i] \\
&\quad \text{for } \$b \text{ in } \text{doc}(“xml doc”)//item \\
&\quad \text{where } \$b$/@ID = \text{CurrentID} \text{ and } \text{TrigRule}(G_x) \\
&\quad \text{return } \text{string}($b$/@ID) \\
\} \)

\(\text{(b)}\)

\(\text{FIG. 7: (a) Algorithm FilterResults. (b) Subroutine RefineSelection.}\)

---

\(^8\) E.g., by analyzing the structure of the SDGraph in Fig. 2, the system can identify implicitly that the user belongs to ethnic group \(E_y\) and national origin \(O_x\), if the user lives in neighborhood \(N_x\) and follows religion \(R_y\).

\(^9\) Assuming that each item in the XML document has an attribute labeled \(ID\).
After result items are filtered, they will be ranked based on the rated scores of the preference vector $D$ of the user’s MDG. The score of an item is defined as the summation of the normalized weights of vector $D$ on the features of the item. Thus, the score $T_x$ of an item is defined as: 

$$T_x = \sum_{i=1}^{N} d_i$$ 

where $d_i \in D$, $D = \{d_1, d_2, \ldots, d_N\}$, and $N$ is the number of weighted features. Items whose scores are high are ranked higher.

We now present example 5 to illustrate Algorithm FilterResults. The example simulates an online grocery system targeting the USA-based SDGs of our running example. The grocery system uses the fragment of XML document shown in Fig. C1 in Appendix C.

Example 5: Consider that a user is looking for canned soup of brand Campbell. SPGProfile would construct an equivalent XQuery query $Q$ as shown in Fig. 8-a. Consider that SPGProfile identified the user’s MDG as $\{E_y, R_x\}$. Line 1 of Algorithm FilterResults would return the following set of canned soups’ IDs: $SItems = \{200015, 200027, 200044, 200058, 200063\}$. Line 3 will call subroutine RefineSelection two times, as follows:

First call: The inputs to the subroutine are SDG $E_y$ and set $SItems$. The WHERE clause will trigger FoodTrigRule($E_y$) (recall Fig. 4). After the substitutions, the subroutine will become as shown in Fig. 8-b. From the IDs in set $SItems$, the subroutine will return the subset: $SItems = \{200027, 200058, 200063\}$.

Second call: The inputs to the subroutine are SDG $R_x$ and set $SItems$. The WHERE clause will trigger FoodTrigRule($R_x$) (recall Fig. 5). After the substitutions, the subroutine will become as shown in Fig. 8-c. From the IDs in set $SItems$, the subroutine will return the subset: $SItems = \{200027, 200058\}$.

<table>
<thead>
<tr>
<th>$Q$:</th>
<th>RefineSelection($E_y$, $SItems$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>for $b$ in doc(&quot;grocery.xml&quot;)/grocery where $b$/Type = “soup” and $b$/brand = “Campbell” return string ($b$/@ID)</td>
<td>1. for ($i = 1; i &gt;= 3; i++$)</td>
</tr>
<tr>
<td></td>
<td>2. CurrentID = $SItems[i]$</td>
</tr>
<tr>
<td></td>
<td>3. for $b$ in doc(&quot;grocery.xml&quot;)/grocery where $b$/@ID = CurrentID and contains($b$/ingredients, “no pork-related products”) return string ($b$/@ID)</td>
</tr>
</tbody>
</table>

**6. System Architecture**

Fig. 9 shows the system architecture. The system administrator inputs an OWL file representing a group profile ontology to module MDGManager. The module converts the OWL ontologies into a SDGraph. Using the structure of the SDGraph, module MDGManager determines MDGs and models them in the form of a MDGraph. Module DBManager identifies the preferences of a SDG from either the preferences of its member users or from published studies. The module will then assign a folder for each SDG in a file system database called SDGs folders. Each folder contains information such as the name and preferences of a SDG, in addition to subfolders. Each subfolder stores information about one of the users belonging to the SDG (such as his preferences). That is, individual subfolders are associated with a SDG folder via file in a file system structure in a two-layer directory tree. Module DBManager keeps track of the number of subfolders within the folder of each SDG $S_{E_x}$. When this number exceeds
a specific threshold, $SE_r$ is activated, and its preferences are stored in database `Triggers Rules` in the form of trigger rule `TrigRule(SE_r)`. Database `Triggers Rules` stores also the XML files. Module `FilterResults` filters results sequentially by applying the preferences of the SDGs to which the user belongs. After each filtering sequence, the temporary results are stored in `Results Buffer`. Module `RankResults` ranks the filtered results, based on the weighted preferences of the user’s MDG.

![System architecture diagram](image)

**FIG. 9: System architecture**

### 7. Experimental Results

We implemented SPGProfile in Java and ran it on an Intel(R) Core(TM)2 Dup CPU processor, with a CPU of 2.1 GHz and 3 GB of RAM, under Windows Vista. We aimed at: (1) simulating online auto and grocery dealers, running the SPGProfile system and targeting 20 USA neighborhoods, (2) obtaining ethnic and religious-based food preference data and region-based car preference data from real individuals living in the 20 neighborhoods, (3) applying the group modeling strategy described previously on the preference data to determine the preferences of the ethnic, religious, and region-based SDGs in the 20 neighborhoods, and (4) comparing the simulated dealers running the SPGProfile system with Google Base [13, 14] and with Oracle XML DB [35].

We selected 20 neighborhoods known for their ethnic and religious diversity. Each neighborhood represents a region-based SDG. From the combined residents of the 20 neighborhoods, those who belong to a same ethnicity represent an ethnic-based SDG, and those who have the same religious beliefs represent a religious-based SDG. We needed to identify some of the residents in the 20 neighborhoods to obtain the preference data. Towards this, we followed the following strategy. We searched the online auto dealer system [9] for used cars offered for sale by owners living in the 20 neighborhoods. By entering a zip code of a neighborhood, [9] returns a list of sellers living in the neighborhood along with their telephone contacts and the features of their cars. We selected some sellers and considered the features and prices of their cars as representative of their preferences on cars. We then phoned and asked them to: (1) rate their preferences on the features of their cars, (2) provide us with their ethnicities and religions, and (3) reveal their preferences and restrictions on canned soup flavors and ingredients.

Along with six volunteers, we contacted the sellers and explained the objectives of our research. Some of them declined to provide us with the information, while others agreed. We stopped the contacts of each neighborhood when the number of successful calls reached 20. Thus, the number of overall successful contacts is 400. We
considered a call successful if the seller revealed his/her ethnicity and rated his/her preferences on cars’ features. From the 400 successful calls, only 278 sellers revealed their religions and canned soup preferences and restrictions. Table 2 shows the number of region, ethnic, and religious SDGs we could construct based on the data we obtained. We constructed 400 car queries and 278 grocery queries based on the data. See Appendix D for a sample of these queries and how they were constructed.

**TABLE 2: Number of region, ethnic, and religious SDGs from which we obtained preference data**

<table>
<thead>
<tr>
<th>State</th>
<th>Minnesota</th>
<th>Texas</th>
<th>Washington</th>
<th>Florida</th>
<th>California</th>
</tr>
</thead>
<tbody>
<tr>
<td>City</td>
<td>Minneapolis</td>
<td>Dallas-Fort Worth</td>
<td>Seattle</td>
<td>Miami</td>
<td>Los Angeles</td>
</tr>
<tr>
<td>Number of neighborhood-based SDGs</td>
<td>5</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>Number of Ethnic-based SDGs</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Number of Religious-based SDGs</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

A demo of the SPGProfile simulating the online auto and grocery dealer systems, which target the 20 neighborhoods, is available at: [http://dbse1.uta.edu/~kamal/?action=home](http://dbse1.uta.edu/~kamal/?action=home). The systems filter and rank results based on the preferences of the 400 subjects living in the 20 neighborhoods. See Appendix E for detailed information about the demo system.

### 7.1 Search Effectiveness Evaluation

**Evaluating Precision on Initially Retrieved Elements:** Users usually want to know only the initially retrieved elements (at lower ranks). We evaluated the search effectiveness of SPGProfile on initially retrieved elements, by comparing it with Google Base and Oracle XML DB. Both search engines do not filter and rank results based on group profiling. Our objective was to have knowledge of SPGProfile’s extent of improvement over the two search engines as a result of considering group profiling. We generated 1000 MBs cars.xml and grocery.xml documents using ToXgene [38]. We ran all of the car and grocery queries against the two documents, using SPGProfile, Google Base, and Oracle XML DB. We computed the Mean Average Precision (MAP) at variable ranks for each of the three systems. For each query submitted to SPGProfile, we revealed to the system some of the SDGs of the subject, for whom the query was constructed. Let \( \{t_1, t_2, ..., t_k\} \in S \) be the set of relevant items for a subject need. Let \( R_{nm} \) be the set of ranked retrieval results from the top result until item \( t_m \). Then, MAP can be computed as shown in Equation 2.

\[
MAP(S) = \frac{1}{|S|} \sum_{n=1}^{S} \frac{1}{k_n} \sum_{m=1}^{k_n} \text{Precision}(R_{nm})
\]

We computed MAP at ranks 5 (top 5 answers), 10, and 15. Tables 3 and 4 show the overall MAPs. As the tables show, SPGProfile achieved much higher precision, especially at rank 5. The tables show also that the MAPs of the three search engines decrease as the rank increases.

**TABLE 3: MAPs using the car queries**

<table>
<thead>
<tr>
<th>Rank</th>
<th>SPGProfile</th>
<th>Google Base</th>
<th>Oracle XML DB</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>0.73</td>
<td>0.27</td>
<td>0.23</td>
</tr>
<tr>
<td>10</td>
<td>0.61</td>
<td>0.18</td>
<td>0.19</td>
</tr>
<tr>
<td>15</td>
<td>0.44</td>
<td>0.11</td>
<td>0.09</td>
</tr>
</tbody>
</table>

\(^{10}\) We asked subjects to rate their preferences on each feature’s characteristic in the range from 1 to 10.

\(^{11}\) Hereinafter we refer to the people we interviewed over the phone as “subjects”.

\(^{12}\) See Appendix D for a sample of the queries and how they were constructed.

\(^{13}\) Recall that each query represents the initial preferences of one of the subjects living in the 20 neighborhoods.

\(^{14}\) The set is constructed based on the weights on item features provided by the subject during a phone interview.

\(^{15}\) The set returned by SPGProfile, Google Base, or Oracle XML DB.
TABLE 4: MAPs using the grocery queries

<table>
<thead>
<tr>
<th>Rank</th>
<th>SPGProfile</th>
<th>Google Base</th>
<th>Oracle XML DB</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>0.68</td>
<td>0.23</td>
<td>0.25</td>
</tr>
<tr>
<td>10</td>
<td>0.57</td>
<td>0.14</td>
<td>0.11</td>
</tr>
<tr>
<td>15</td>
<td>0.34</td>
<td>0.10</td>
<td>0.07</td>
</tr>
</tbody>
</table>

**User Evaluation of Search Effectiveness:** We asked 18 Computer Science PhD and MS students at the University of Texas-Arlington to evaluate the SPGProfile grocery demo system. The students belong to two different ethnic backgrounds and three ancestry origins. Some of them consider religion to be irrelevant and the others follow two different religions. We asked each of them to give us two lists of items. The first contained a list of 10 canned food items (from the brands used by SPGProfile) ranked based on the subject’s own preferences. The second contained the top 10 canned food items ranked by SPGProfile based on the preferences of the subject’s ethnic, religious, and/or national origin groups. We then measured the average distance between the two lists using the Euclidean distance measure shown in Equation 3.

\[
d(\sigma_s, \sigma_r) = \sum_{x \in X} |\sigma_s(x) - \sigma_r(x)|
\]  

(3)

\[\sigma_s \in [0,1]^{[X]}\] - the list of items ranked by subject \( u \).

\[\sigma_r \in [0,1]^{[X]}\] - the list of items ranked by SPGProfile.

\( X \) - the set of items.

\( \sigma_s(x) \) and \( \sigma_r(x) \) - the position of item \( x \in X \) in the lists \( \sigma_s \) and \( \sigma_r \) respectively (a ranking of a set of \( n \) items is represented as a permutation of the integers 1, 2, \ldots, \( n \)).

Fig. 10 shows the distances. The average distance is 12.5, which indicates “closeness” between the lists ranked by the subjects and the lists ranked by the SPGProfile system.

**7.2 Search Efficiency Evaluation**

We evaluated the overhead of filtering and ranking results endured by SPGProfile by comparing its search efficiency with Google Base and Oracle XML DB. We ran each of the 400 car queries against cars.xml documents of 8 different sizes using the three search engines. We also ran each of the 278 grocery queries against grocery.xml...
documents of 8 different sizes using the three search engines. The sizes of the documents ranged from 10-1000 MB. We computed the average query execution time under each different document size. Fig.11 shows the results. As the figure shows, the average query execution time of SPGProfile ranged from 1.1-1.2 times the execution time of an equivalent Google Base query, and ranged from 1.1-1.6 times the execution time of an equivalent Oracle XML DB query. Thus, the overhead of filtering and ranking results endured by SPGProfile is not very expensive.

![Fig. 11: Average query execution time under variable document sizes](image)

7.3 Comparing Preferences of Users with Preferences Published in Studies

We aim at determining which of the following two approaches gives ranked lists of items closest to lists ranked based on the preferences of the subjects living in the 20 neighborhoods: (1) identifying the preferences of SDGs from the preferences of its member users, and (2) identifying the preferences of SDGs from preference data obtained from published studies. We made two copies of SPGProfile. Copy 1 employs the approach described in section 4.1. Copy 2 employs the approach described in section 4.2. Copy 2 uses preference data about SDGs obtained from [20, 2, 39, 20, 33, 12, 5, 24]. We used the cosine-similarity measure in Equation 4:

$$
sim(D_{u_m}, D_{v_i}) = \frac{\sum_{i \in P}(r_{u_m,i} - \bar{r}_{u_m})(r_{v_i,i} - \bar{r}_{v_i})}{\sqrt{\sum_{i \in P}(r_{u_m,i} - \bar{r}_{u_m})^2} \sqrt{\sum_{i \in P}(r_{v_i,i} - \bar{r}_{v_i})^2}}
$$

(4)

$D_{u_m}$ - the preference vector of subject $u_m$.
$D_{v_i}$ - the preference vector of the subject’s MDG $v_i$.
$r_{u_m,i}$ and $r_{v_i,i}$ - the preference weight on feature characteristic $i$ rated by $u_m$ and $v_i$ respectively.
$\bar{r}_{u_m}$ - the mean preference weight on features’ characteristics rated by $u_m$ and co-rated by $v_i$.
$\bar{r}_{v_i}$ - the mean preference weight on features’ characteristics rated by $v_i$ and co-rated by $u_m$.
$P = I_{u_m} \cup I_{v_i}$
$I_{u_m}$ and $I_{v_i}$ - the set of features’ characteristics rated by $u_m$ and $v_i$ respectively.
We measured the similarity between the preference vector of each subject and: (1) the preference vector of the subject’s MDG determined by copy 1, and (2) the preference vector of the subject’s MDG determined by copy 2. Fig. 12 shows the similarity results using car preference vectors and Fig. 13 shows the results using grocery preference vectors. As the figures show, as the number of accumulated subjects’ vectors increases, the similarity between them and the vectors of the subject’s MDG determined by copy 1 increases. This finding supports the known hypothesis that the more members of a group are considered the more reflective their common preferences are of the preferences of each member. The figures show also that the similarity between the subjects’ vectors and the vectors of the subjects’ MDGs determined by copy 2 is independent of the number of accumulated subjects’ vectors. We can conclude that approach 1 is better than approach 2, especially when the number of users is relatively high.

**FIG. 12: Similarity using cars preference vectors**

**FIG. 13: Similarity using grocery preference vectors**
8. Previous Studies and Conclusion

With the growth of massive information on the Web, it has become increasingly difficult to search for useful information. As one of the most promising approaches to alleviate this overload, recommender systems have emerged in domains such as E-commerce, digital libraries, and knowledge management. In general, recommendation systems suggest items or products by analyzing what users with similar tastes have chosen in the past [19]. There are two prevalent approaches to formulate recommendations: collaborative filtering recommendation and content-based recommendation. They depend on the type of items to be recommended and on the way that user models [1, 16] are constructed.

**Collaborative Filtering-Based Recommendation:** In collaborative filtering approach [11, 18, 19, 17], information is filtered for a larger group of users. The term collaborative filtering was coined by Goldberg et al. [15]. Collaborative-filtering algorithms aim to identify users that have relevant interests and preferences by calculating similarities and dissimilarities between user profiles. The idea behind this method is that it may be of benefit to one’s search for information to consult the behavior of other users who share the same or relevant interests.

**Content-Based Recommendation:** A content-based approach [6, 23, 31] provides recommendations by comparing an item’s content with the content that the user is interested in. In this approach, a model of user ratings is first developed. Then the filtering process is envisioned by computing the expected value of a user prediction given the user’s ratings on other items. Content-based algorithms are principally used when documents are to be recommended, such as web pages, publications, or news. The agent maintains information about user preferences either by initial input about user’s interests during the registration process or by rating documents. Recommendations are formed by taking into account the content of documents and by filtering in the ones that better match the user’s preferences and logged profile.

Common interactions that take place in a typical recommendation system include ratings, transactions, feedback data etc. Most systems use one of the following for the acquisition of user knowledge and preferences. Some approaches [26] first model and gather user's search history to construct a user profile and then construct a general profile based on the ODP [34] category hierarchy. Other systems ask users for their preferences explicitly as the main post-query method for automatically improving the systems’ accuracy of users’ need. Alternatively, they may use implicit feedback techniques. Implicit feedback techniques unobtrusively draw usage data by tracking and monitoring user behavior without explicit user involvement. Systems such as [22, 43] obtain data by exploiting previous search history. Other approaches use machine learning to analyze user data [44].

To the best of our knowledge, inferring a user profile from the profiles of groups defined based on ethnic, cultural, religious, etc. has not been researched in any recommendation-based study. It is an area that could further simplify personalized search and enhance search results. In this paper, we proposed an XML-based recommender system, called SPGProfile. It is a type of collaborative information filtering system, and is based on the combination of search-by-query and recommendations. SPGProfile can be used for various practical applications, such as Internet or other businesses that market preference-driven products. We presented a novel approach to XML search that leverages group information to return more relevant query answers for users. The proposed approach simplifies the personalization process by: (1) pre-defining and identifying the preferences of various categories of groups, and (2) filtering and ranking results based on the preferences of the groups to which the user belongs. We experimentally evaluated the search effectiveness and efficiency of SPGProfile by comparing it with Google Base [13, 14] and with Oracle XML DB [35]. The experimental results showed that SPGProfile achieved much higher precision than the two search engines on initially retrieved elements. The experimental results showed also that the average query execution time of SPGProfile ranged from 1.1-1.2 times the execution time of an equivalent Google Base query, and ranged from 1.1-1.6 times the execution time of an equivalent Oracle XML DB query. These results are indicative that the overhead endured by SPGProfile for filtering and ranking results is not very expensive. We can conclude that group profiling can be an effective and efficient retrieval mechanism.
References

[38] ToXgene (2005). ToXgene - the ToX XML Data Generator
Appendix A

Fig. A shows a group profile ontology in the form of OWL ontology. The ontology defines the relations between the USA-based SDGs shown in the SDGraph in Fig. 2.

FIG. A: OWL ontology representing USA-based group profile ontology
Appendix B

Fig. B shows Algorithm ConstructMDGraph, which identifies all possible MDGs that exist because of the interrelations between SDGs. The input to the Algorithm is a SDGraph represented by the adjacency-list Adj. The algorithm works as follows. The elements of each set initialized in line 1 are MDGs located in the same hierarchical level of a MDGraph. Lines 2 - 5: If SDGs v and u are adjacent in the SDGraph (v ∈ Adj[u]), their intersection forms a MDG, which is located in hierarchical level 1 of the MDGraph. Lines 6-11: Function DiffDom takes two SDGs as input and returns true if they belong to different domains. If MDGs V_i, V_j ∈ level_1 share a common SDG, and the rest of the SDGs forming them have different domains, then their intersection forms a MDG in level 2. Lines 12-18: lines 6-11 are repeated with the consideration that V_i, V_j ∈ level_{2+k} (where k = 0, 1,...) until no more MDG can be formed.

```
ConstructMDGraph { 
1. Initialize to null sets level_1, level_2, ...
2. for each two SDGs v, u ∈ Adj { //Adj is an adjacency list representing a SDGraph
3.         if (v ∈ Adj[u])
4.             Insert MDG { v ∩ u } in set level_1
5.         } //end for
6. for each two MDGs V_i, V_j ∈ level_1 {
7.         if (there exist a SDG S ∈ V_i, V_j)
8.             if (each v ∈ V_i, u ∈ V_j (where v, u ≠ S) DiffDom (v, u) )
9.                 Insert MDG { V_i ∩ V_j } in set level_2
10. } //end for
11. } //end for
12. k → 0
13. while a new MDG can be formed {
14.         Repeat lines 6-11 with the consideration that V_i, V_j ∈ level_{2+k}
15.         k = k + 1
16.         Store the resulting MDG in set level_{2+k}
17. } //end while
18. } //end of the algorithm
```

FIG. B: Algorithm ConstructMDGraph
Appendix C

Figures C1 and C2 show fragments of XML documents used by our simulated auto and grocery dealer systems.

```
<groceries>
  <grocery ID = “200004”>
    <Type> soup </Type>
    <brand> Progresso </brand>
    <flavor> spicy </flavor>
    <ingredients> vegetables, rice, pork </ingredients>
  </grocery>
  <grocery ID = “200015”>
    <Type> soup </Type>
    <brand> Campbell </brand>
    <flavor> regular </flavor>
    <ingredients> vegetables, rice, no pork-related products </ingredients>
  </grocery>
  <grocery ID = “200027”>
    <Type> soup </Type>
    <brand> Campbell </brand>
    <flavor> spicy </flavor>
    <ingredients> vegetables, rice, no pork-related products </ingredients>
  </grocery>
  <grocery ID = “200044”>
    <Type> soup </Type>
    <brand> Campbell </brand>
    <flavor> regular </flavor>
    <ingredients> Chicken Flavor, Salt, Sugar, Onion </ingredients>
  </grocery>
  <grocery ID = “200058”>
    <Type> soup </Type>
    <brand> Campbell </brand>
    <flavor> spicy </flavor>
    <ingredients> Beef Stock, Water, Enriched Egg, Noodles, no pork-related products, Cooked Beef, Tomato Puree </ingredients>
  </grocery>
  <grocery ID = “200063”>
    <Type> soup </Type>
    <brand> Campbell </brand>
    <flavor> spicy </flavor>
    <ingredients> Beans, Bacon, Tomato, Puree, Salt </ingredients>
  </grocery>
</groceries>

<cars>
  <car ID = 10520>
    <make> Ford </make>
    <body-style> SUV </body-style>
    <color> black </color>
    <year> 2007 </year>
    <price> 19000 </price>
    <fuel-type> ethanol flexible-fuel </fuel-type>
    <MPG> 34 </MPG>
    <safety-feature> ABS brakes </safety-feature>
    <wheel-drive> 4WD </wheel-drive>
  </car>
  <car ID = 10884>
    <make> Chevrolet </make>
    <body-style> SUV </body-style>
    <color> red </color>
    <year> 2008 </year>
    <price> 21000 </price>
    <fuel-type> gasoline </fuel-type>
    <MPG> 34 </MPG>
    <safety-feature> automatic break </safety-feature>
    <wheel-drive> RWD </wheel-drive>
  </car>
  <car ID = 11134>
    <make> Dodge </make>
    <body-style> SUV </body-style>
    <color> black </color>
    <year> 2005 </year>
    <price> 17000 </price>
    <fuel-type> diesel </fuel-type>
    <MPG> 24 </MPG>
    <safety-feature> ESC </safety-feature>
    <wheel-drive> FWD </wheel-drive>
  </car>
  <car ID = 101250>
    <make> Audi </make>
    <body-style> SUV </body-style>
    <color> silver </color>
    <year> 2006 </year>
    <price> 23000 </price>
    <fuel-type> hybrid-electric </fuel-type>
    <MPG> 32 </MPG>
    <safety-feature> ESC </safety-feature>
    <wheel-drive> RWD </wheel-drive>
  </car>
</cars>
```

FIG. C1: A fragment of groceries.xml document

FIG. C2: A fragment of cars.xml document
Appendix D

Car queries

Table D1 shows basic features of cars offered for sale belonging to a sample of two subjects living in the 20 neighborhoods. The table shows also the weights on features’ characteristics provided by the two subjects during our phone interviews. Table D2 shows queries constructed from the basic features in Table D1 simulating the subjects’ initial preferences. We constructed 400 car queries in the same manner to simulate the initial preferences of 400 subjects living in the 20 neighborhoods. The weights in Table D1 are used for determining the preferences of the subjects’ SDGs, using the group modeling strategy described in section 4.1.

TABLE D1: Basic features of cars offered for sale belonging to two subjects along with the subjects’ weight vectors

<table>
<thead>
<tr>
<th>Subject #</th>
<th>Basic features of the car offered for sale belonging to the subject</th>
<th>Subject’s weights on features’ characteristics ranged from 1-10</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Make</td>
<td>Body-style</td>
</tr>
<tr>
<td>1</td>
<td>Ford</td>
<td>SUV</td>
</tr>
<tr>
<td>2</td>
<td>Dodge</td>
<td>Sedan</td>
</tr>
</tbody>
</table>

TABLE D2: Query constructed from the cars’ basic features in Table D1 to simulate the initial preferences of the subjects

<table>
<thead>
<tr>
<th>Subject #</th>
<th>Subject's query</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Looking for a black Ford SUV made in or after 2007</td>
</tr>
<tr>
<td>2</td>
<td>Looking for a gray Dodge sedan made in or after 2002</td>
</tr>
</tbody>
</table>

Grocery queries

Table D3 shows the type of information we collected from the subjects for our grocery system (e.g., group affiliations and weights on characteristics of canned food features). The Table shows also how we constructed queries simulating the subject’s initial preferences from basic features of canned food. We constructed 278 grocery queries in the same manner to simulate the initial preferences of 278 subjects living in the 20 neighborhoods.

TABLE D3: The type of information collected from the subjects for the grocery system and the queries simulating their initial preferences

<table>
<thead>
<tr>
<th>Subject #</th>
<th>Ethnicity</th>
<th>Ancestry Origin</th>
<th>Religion</th>
<th>Subject’s weights on features’ characteristics ranged from 1-10.</th>
<th>Query constructed from basic features of canned food to simulate the subject’s initial preferences</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>(8, 10, 4) Chili pepper no meat rice starch</td>
<td>Looking for 15.25 ounce canned soup brand Healthy Choice</td>
</tr>
<tr>
<td>2</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>(5, 10) Organic spices no pork</td>
<td>Looking for 10.75 ounce canned soup brand Progresso</td>
</tr>
</tbody>
</table>
Appendix E

The demo simulates an online auto dealer and online grocery dealer systems, running on 1000 MB data. The systems target 20 US neighborhoods, from which we interviewed subjects to obtain preference data.

The grocery system

It stores preference trigger rules for the ethnic, religious, and national origin SDGs, to which the subjects belong. The preferences of each SDG were determined using the group modeling strategy described in section 4.1. The user has the option to select from drop-down menus his ethnic, national origin, and/or religious SDGs. By selecting a SDG, the system triggers the SDG’s trigger rules to filter and rank the user’s initial results. By selecting certain combinations of zip codes and SDGs, the system can identify implicitly other SDGs, to which the user belongs, by analyzing the structure of the SDGraph.

The auto system

It stores car preference trigger rules for the region-based SDGs, to which the 400 subjects belong. By selecting a zip code from a drop-down menu, the system triggers the trigger rules of the neighborhood and region in which the user lives. The triggers filter the user’s results based on the preferences of his neighborhood and region. Neighborhoods’ preferences include car prices, which are governed, mostly, by the neighborhoods’ median incomes. Regions’ preferences include weather-related car features, and fuel-efficiency features influenced by local governments’ regulations and incentives.

Preferences of the groups, which are defined based on ethnicity or national origin, involve food flavor preferences and/or ingredient content preferences or restrictions. Preferences of the groups, which are defined based on religion, involve food restrictions dictated by religious teachings e.g., restriction of beef, restriction of pork, etc.