Group Preferences in Social Network Services

Florian Wenzel
Institute for Computer Science
University of Augsburg
86135 Augsburg, Germany
wenzel@informatik.uni-augsburg.de

Werner Kießling
Institute for Computer Science
University of Augsburg
86135 Augsburg, Germany
kiessling@informatik.uni-augsburg.de

ABSTRACT

With the beginning of the new millenium, the concept of group interactions in communication systems was boosted by the emergence of Web 2.0 technologies. Based on this new area of application, the notion of group decisions and group preferences also evolved, leading to new requirements for corresponding modeling frameworks. Purely numeric approaches are barely able to meet these newly emerging challenges. Therefore, we provide a comprehensive group preference framework to overcome the deficits of previous solutions and demonstrate possible applications in social network services. The concept provides both numeric and semantic means which can be applied to determine group preferences and to perform further evaluations based on the semantic value of preference terms. With Preference SQL a powerful system exists to implement the presented group preference model using standard commercial databases.

1. INTRODUCTION

With the beginning of the new millenium, the concept of group interactions in communication systems was boosted by the emergence of Web 2.0 technologies. From social networking sites to e-commerce platforms, communities have been established allowing users to provide recommendations for products or to share common interests. Consequently, the next step in this development is to combine these single user valuations into consensus decisions for a group. Lately, a new method has been introduced to combine single user ratings into a common group recommendation [1]. In the field of operations research, determining a group decision among alternatives with multiple attributes is also a well-known problem that has been investigated for decades. Some of these approaches even introduce database-aided algorithms [6, 10]. However, what all these former attempts have in common is that they approach the topic from a purely numeric point of view by using utility functions or ranking techniques [7] for single user opinions and additional numeric means to express disagreement between users or the fuzziness of user preferences [2, 8].

GvD Workshop'10, 25.-28.05.2010, Bad Helmstedt, Germany.

In the course of this paper we therefore want to introduce an augmented approach to tackle the problem of group preferences. The preference framework introduced in [3] and extended in [4] provides powerful methods to model single user preferences and to combine the corresponding preference terms to form a common group preference. Furthermore, preferences can be interpreted in both a numeric and even more importantly a semantic fashion. These facts allow for the introduction of a numeric quality measure for group preferences and the use of semantics as background knowledge for heuristics which can be applied to improve that quality. With Preference SQL, a powerful database-aided implementation exists that makes the presented findings ready to use in specific application contexts.
The remainder of this paper is structured as follows: section 2 introduces means to model single user preferences on multiple attributes and describes how these base preferences can be combined to form complex user preferences. The essential questions of how user preference terms influence a common group preference and how user hierarchies can be introduced into the group are discussed. Section 3 expands into the details of group preference evaluation and depicts the difference between the numeric and the semantic meaning of corresponding preference terms. These formal demonstrations are further illustrated in a use case scenario staging a fictious social network service for outdoor enthusiasts in section 4. The implications of the described findings for future research are highlighted in section 5 followed by concluding thoughts.

2. GROUP PREFERENCE FORMATION

Numeric approaches often determine an implicit utility function based on previous decisions of a group member in order to determine group member preferences. This generalization of past preferences has to be assessed critically since environmental aspects and moderator variables [9] that vary across situations are neglected and thus these functions not necessarily reflect the individual's intuitive preferences at the given moment. In contrast, [3, 4] introduce a hierarchy of base preference constructors that provide a framework to express preferences on single attributes. This process can be seen as an explicit means for a group member to express opinions on multiple criteria. These base preferences can further be combined into a complex single user preference term with the help of constructors that allow to value some base preferences more than others. Finally, user preference terms are joined into a single group preference by the same operators that help to form complex single user preferences.

2.1 Expression of Group Member Preferences

A preference P of the kind "I like y more than x" is formally defined as $x <_P y$. This intuitive definition of preferences in terms of "better-than" has a natural counterpart in mathematics, namely strict partial orders. Besides basic "better-than" preferences, often complex valuations have to be expressed, which requires the accumulation of base preferences into more complex ones. This leads to the inductive construction of preferences $P = (A, <_P)$ with the help of complex preference constructors. P is accordingly specified by a preference term wich fixes the attribute names A and the strict partial order $<_P$. Like base preferences, preference terms that consist of complex preference constructors also respresent strict partial orders, therefore laying the foundation for a preference algebra [3].

Group members express their preferences on single attributes which can have either numerical or non-numerical domains. Numeric preferences include amongst others:

- LOWEST(A): $x <_P y$ iff $x > y$
- HIGHEST(A): $x <_P y$ iff $x < y$
- AROUND(A,z): $x <_P y$ iff $|x - z| > |y - z|$
- SCORE(A,f): $x <_P y$ iff $f(x) < f(y)$ for a function f

Non-numerical preferences determine likes or dislikes on categoric attribute values and are e.g. expressed by:

- POS(A, *pos*): $x <_P y$ iff $x \notin pos \land y \in pos$, with *pos* being a set of preferred values
- NEG(A,*neg*): $x <_P y$ iff $y \notin neg \land x \in neg$, with *neg* being a set of disliked values

Each group member specifies base preferences on some or all attributes $A_1, \ldots, A_n$ of a relation R. These base preferences are afterwards combined using complex preference constructors such as:

- Pareto $P_1 \otimes P_2$: equal importance of preferences P_1 and P_2
- Prioritization $P_1 \& P_2$: preference P_1 is more important than P_2

The Pareto constructor stands for equal importance of participating preferences. In contrast, a prioritization constructor expresses a favor for the preference provided as first attribute. Only if two tuples are equally important concerning the first preference then the second preference is used for further discrimination.

As a result of this first step towards a common group preference, a complex preference term for each group member in preference algebra is formed. In a second step, these terms $T_1, \ldots, T_k$ of k single members are combined into a single group preference.

2.2 Formation of a Group Preference Term

Given k single user preference terms $T_1, \ldots, T_k$ there are multiple ways to form a group preference by using the complex preference constructors described before. A group hierarchy can be introduced by the use of Pareto and prioritization constructors. A flat hierarchy in which all group members are of equal value is defined by a group preference

$$P_{group} = T_1 \otimes \ldots \otimes T_k. \tag{1}$$

Accordingly, some group members can be marked as more important than others by the use of the prioritization constructor, e.g.

$$P_{group} = T_1 \& T_2 \& \ldots \& T_k. \tag{2}$$

In this case, group member one is more important than group member two who is more important than group member three and so forth.
Most likely, these two possibilities are combined to form an individual group hierarchy, e.g.

$$P_{group} = T_1 \& (T_2 \otimes \ldots \otimes T_k). \tag{3}$$

Especially hierarchy patterns depicted in equations 1 and 3 frequently occur, with the first equation describing a group with no hierarchy and the third equation a group with one leader amongst otherwise equal members. However, virtually all combinations of single user terms are possible, e.g.

$$P_{group} = T_1 \& (T_2 \otimes T_3 \otimes T_{k-2} \& (T_{k-1} \otimes T_k)). \tag{4}$$

Equivalent to single users preferences, group preferences can be transformed to statements in Preference SQL for further evaluation.

2.3 Implementation with PSQL

The Preference SQL (PSQL) system implements the SQL 92 standard and additionally supports a *PREFERRING* clause which allows the expression of soft constraints. Hence, it provides a powerful means to implement group preferences in present database systems. A user preference term in preference algebra can be transformed into an equivalent PSQL statement which is evaluated by PSQL. The result of such a preference query for preference P on relation R is denoted as $\sigma[P](R)$. Preference queries are evaluated according to a Best-Matches-Only strategy (BMO) which means that a perfect match is returned if such tuples exist and otherwise the best matches are retrieved but nothing worse. This proceeding behaves contrary to conventional SQL queries which determine only exact matches. As a consequence, preference queries represent soft constraints that should be fulfilled while SQL queries are interpreted as hard constraints that have to be matched without exception. A PSQL statement is similar to a SQL statement, with the distinctive feature of an additional *PREFERRING* clause:

SELECT <*attributes*> *FROM* <*table reference*>
WHERE <*hard constraints*>
PREFERRRING <*soft constraints*>

Within the *PREFERRING* clause, POS preferences are indicated by the keyword *IN* followed by the set of preferred values while NEG preferences are indicated by *NOT IN* and the set of unliked values. AROUND and BETWEEN preferences are obviously stated by the corresponding keywords *AROUND* and *BETWEEN* in combination with a numeric value or interval. Finally, the Pareto constructor is described by the keyword *AND* while *PRIOR TO* stands for a prioritization.

While an extended documentation of the PSQL syntax can be found in [5], the following example depicts a user preference on a relation R and the equivalent PSQL statement:

$$P_{U_1} = (POS(A_1, \{dif\}) \otimes NEG(A_2, \{lsc\}))\& \\ (AROUND(A_3, 5) \otimes BETWEEN(A_4, [2, 3])) \quad (5)$$

*SELECT * FROM R*
PREFERRING (A_1 IN ('dif')
AND A_2 NOT IN ('lsc'))
PRIOR TO (A_3 AROUND 5
AND A_4 BETWEEN 2 AND 3)

3. GROUP PREFERENCE EVALUATION

Once a group preference is formed and the corresponding PSQL statement determined, a PSQL query returns the BMO set for the specified group preference. This result represents more than just the intersection of single user preferences as the following two examples based on tuples representing hiking tours illustrate:

id	*dif*	*lsc*
1	easy	nice
2	medium	sparse
3	hard	neutral
4	easy	special
5	medium	special

Table 1: Excerpt of a tour database

Example 1:
Based on the tuples depicted in Table 1, users U_1 and U_2 state the following preferences on attributes *difficulty (dif)* and *landscape (lsc)*:

- $P_{U_1} = POS(dif, \{easy\}) \otimes POS(lsc, \{nice\})$
- $P_{U_2} = POS(dif, \{medium\} \otimes POS(lsc, \{special\})$

In this example it can easily be seen that user U_1 favors the tuple with id=1 while U_2 prefers the tuple with id=5. The result $\sigma[P_{U_1} \otimes P_{U_2}](R)$ of the common group preference $P_{group} = P_{U_1} \otimes P_{U_2}$ is depicted in Table 2.

id	*dif*	*lsc*
1	easy	nice
4	easy	special
5	medium	special

Table 2: BMO set of group preference $P_{U_1} \otimes P_{U_2}$

Based on the results of single user preferences $\sigma[P_{U_1}](R)$ for user U_1 and $\sigma[P_{U_2}](R)$ for U_2 this evaluation shows that the group preference contains more than just the aggregation of single user preferences $P_\cup = \sigma[P_{U_1}](R) \cup \sigma[P_{U_2}](R)$ or the intersection $P_\cap = \sigma[P_{U_1}](R) \cap \sigma[P_{U_2}](R)$. Instead, compromises are found such as the tour with id=4.

Example 2:
This effect becomes even more obvious if no perfect matches exist for single users:

- $P_{U_1} = POS(dif, \{easy\}) \otimes POS(lsc, \{sparse\})$
- $P_{U_2} = NEG(dif, \{hard\}) \otimes NEG(lsc, \{nice\}$

In this case, no perfect match exists in R for both U_1 and U_2. The corresponding BMO sets for U_1 and U_2 are listed in Tables 3 and 4.

id	*dif*	*lsc*
1	easy	nice
2	medium	sparse
4	easy	special

Table 3: BMO set $\sigma[P_{U_1}](R)$ for U_1

id	*dif*	*lsc*
2	medium	sparse
4	easy	special
5	medium	special

Table 4: BMO set $\sigma[P_{U_2}](R)$for U_2

Eventually, Table 5 shows the result $\sigma[P_{U_1} \otimes P_{U_2}](R)$ of the combined group preference $P_{group} = P_{U_1} \otimes P_{U_2}$ for U_1 and U_2 which in this case is the intersection between the two single user preferences. However, example 1 already showed that this is not the case in general.

id	*dif*	*lsc*
2	medium	sparse
4	easy	special

Table 5: BMO set of group preference $P_{U_1} \otimes P_{U_2}$

The two examples illustrate the behavior of group preferences consisting solely of base preferences and Pareto constructors. Correspondingly, the same results can be shown for group preferences using prioritization constructors and group terms consisting of a mixture of complex Pareto and prioritization preference terms. Furthermore, different preferences on the same attribute have been treated equally with respect to any other preference constellation in this evaluation. Considering the fact that semantically opposite preferences on the same attribute stated by different members of a group might end up in a common group preference, it becomes obvious that an approach disregarding preference semantics would produce poor group preference results. The concluding outlook therefore outlines how the special semantics of these pairings can be used as indicator for the formation of subgroups.
To illustrate the presented findings, a use case scenario is evaluated in the following section.

4. A SOCIAL NETWORK SERVICE FOR OUTDOOR ACTIVITIES

The formal foundations of previous sections are now all put to action in a fictionary social network service. In this scenario, a website offers a database of tours for various outdoor activities in which a user can find suitable entries by specifying desired tour features in a predefined search overview. Furthermore, a community exists in which every user keeps a profile. This profile can be used to store search results and to create personal tour entries which can be shared with other members of the community.

This pre-existing service is now augmented by the introduction of preferences into the search process. A user states preferences directly in the personal profile of the community. This process doesn't have to be performed repeatedly and can be assisted by suitable GUI components for each preference, such as sliders for AROUND preferences or checkbox groups for POS or NEG preferences. These preferences are then transformed into equivalent PSQL statements on a relation containing the tour entries. Considering a user U_1 and four tour attributes named *difficulty*, *distance*, *duration* and *landscape* in a relation called *hiking*, a preference for a distance close to 50 km, an easy difficulty level, a duration between five and six hours and a dislike for sparse landscapes can be expressed by the following PSQL statement:

*SELECT * FROM hiking*
PREFERRING (dist ARROUND 50)
    AND dif IN ('easy')
    AND (dur BETWEEN 5 AND 6)
    AND lsc NOT IN ('sparse');

This statement can be used any time the user whishes a tour suggestion and can be extended by hard constraints to restrict potential candidates to specific locations. In this case, if the relation also contains an attribute *region*, a corresponding request with restriction to tours in Franconia would be:

*SELECT * FROM hiking*
WHERE region = 'franconia'
PREFERRING (dist ARROUND 50)
    AND dif IN ('easy')
    AND (dur BETWEEN 5 AND 6)
    AND lsc NOT IN ('sparse');

This example of applying single user preferences in individualized search processes clearly highlights the advantages of preferences in contrast to conventional approaches via search masks and SQL. Using the conventional approach, the user has to state the same search criteria over and over again, even if user preferences remain constant. With the preference approach, preferences statements are only changed if the user preferences indeed evolve. Furthermore, PSQL results deliver BMO set which means that empty result and flooding effects are avoided. Conventional approaches, in contrast, show no result tolerance and thus don't list tours that have distances of 52 or 49 kilometers if a distance of 50 kilometers is specified in the search statement. In this case, the user is required to frequently reformulate search requests to obtain suitable results.

Now consider the group function that is common to most social networking websites which allows users to join groups based on common interests. In this case, preferences provide a major improvement for the social network service. Once a user joins a group, e.g. the group representing the local hiking club, common group activities can be organized based on single user preferences. Considering PSQL conform preference terms $T_1, \ldots, T_k$ for k group members, a common PSQL query is constructed as follows:

*SELECT * FROM hiking*
PREFERRING (T_1 *AND* $\ldots$ *AND* T_k)*;*

This presented statement reflects a flat group hierarchy. Furthermore, a group administrator might know of further restrictions that have to be imposed onto the group decision, e.g. limitations for the difficulty of the tour. These preference hierarchies can be integrated seamlessly:

*SELECT * FROM hiking*
PREFERRING dif IN ('easy')
    PRIOR TO (T_1 *AND* $\ldots$ *AND* T_k)*;*

Finally, administrators may form subgroups by the definition of critical attribute values, e.g. to define a beginner and an advanced hiking group:

*SELECT * FROM hiking*
PREFERRING dif IN ('easy')
    PRIOR TO (T_1 *AND* $\ldots$ *AND* T_k)*;*

*SELECT * FROM hiking*
PREFERRING dif NOT in ('easy')
    PRIOR TO (T_1 *AND* $\ldots$ *AND* T_k)*;*

After these groups are determined, single users have to be assigned to groups based on a quality metric that determines how good the user fits into a particular group.

5. SUMMARY AND OUTLOOK

Despite past activities of various disciplines in the exploration of group decisions, newly emerged areas of application demand an extended framework to model the complex process of group preference determination. This procedure of merging single user preferences into a group preference consequently raises the question of how well this group consensus represents the notion of every single group member. In contrast to group decision support systems which are constructed to force a consensus among members, social network services act upon the maxime that connections are formed freely by users who share common interests. While this assumption is certainly true with regard to some aspects, interests might yet differ in other parts of a specific domain. In the use case described in section 4, users built groups based on a common interest in outdoor activities or the membership in a hiking club. Nevertheless, essential differences might occur based on diverse ability levels, e.g. in terms of preferred tour difficulty. In these cases it might be

more fortunate to form multiple subgroups instead of impose a single group on all members.
The presented preference framework provides various new insights to approach this problem. Preferences can be used to define a quality measure which in turn determines the homogeneity of a group by measuring the quality of its group preference. In case the measure is below a certain threshold, the semantics of preferences further allow to detect certain phenomena that lead to a heterogeneous outcome. This analysis finally delivers starting-points for the formation of subgroups.

Being able to explicitly state preferences in a community profile certainly is an additional benefit of the presented approach, however, it requires an extension of already installed social network services. For present networking sites a preference mining subservice might be of special interest that generates base preferences out of pre-existing text blocks.

The presented approach provides a set of means which extend the possibilities of modeling group preference far beyond the status quo of numeric techniques and points out starting-points for new research efforts. The introduction of semantics into the group preference model provides an additional layer that can be used as background knowledge for informed heuristics. Most recent projects at the University of Augsburg include the definition of quality metrics for group preferences and the development of heuristics to improve that group quality dynamically. Future publications will address some of these newly emerged aspects in detail.

6. REFERENCES

[1] S. Amer-Yahia, S. B. Roy, A. Chawla, G. Das, and C. Yu. Group Recommendation: Semantics and Efficiency. *PVLDB*, 2(1):754–765, 2009.

[2] H.-M. Hsu and C.-T. Chen. Aggregation of fuzzy opinions under group decision making. *Fuzzy Sets and Systems*, 79(3):279 – 285, 1996.

[3] W. Kießling. Foundations of Preferences in Database Systems. In *VLDB '02: Proceedings of the 28th international conference on Very Large Data Bases, Hong Kong, China*, pages 311–322. Morgan Kaufmann, 2002.

[4] W. Kießling. Preference Queries with SV-Semantics. In J. R. Haritsa and T. M. Vijayaraman, editors, *COMAD '05: Advances in Data Management 2005, Proceedings of the 11th International Conference on Management of Data, Goa, India*, pages 15–26. Computer Society of India, 2005.

[5] W. Kießling and G. Köstler. Preference SQL - Design, Implementation, Experiences. In *VLDB '02: Proceedings of the 28th international conference on Very Large Data Bases, Hong Kong, China*, pages 990–1001. Morgan Kaufmann, 2002.

[6] S. H. Kim, S. H. Choi, and B. S. Ahn. Interactive group decision process with evolutionary database. *Decision Support Systems*, 23(4):333 – 345, 1998.

[7] O. K. Ngwenyama and N. Bryson. Eliciting and mapping qualitative preferences to numeric rankings in group decision making. *European Journal of Operational Research*, 116(3):487 – 497, 1999.

[8] H. Nurmi. Approaches to collective decision making with fuzzy preference relations. *Fuzzy Sets and Systems*, 6(3):249 – 259, 1981.

[9] S. Sharma, R. M. Durand, and O. Gur-Arie. Identification and Analysis of Moderator Variables. *Journal of Marketing Research*, 18(3):291–300, 1981.

[10] R. Vetschera. Integrating databases and preference evaluations in group decision support : A feedback-oriented approach. *Decision Support Systems*, 7(1):67 – 77, 1991.