Rules for Using Multi-Attribute Utility Theory for Estimating a User's Interests

Ralph Schäfer[1]

DFKI GmbH, Stuhlsatzenhausweg 3, 66123 Saarbrücken
Ralph.Schaefer@dfki.de

Abstract. In this paper, we show that Multi-Attribute Utility Theory (MAUT), a prescription for evaluating objects, can be ascribed as evaluation process to a user when estimating the user's interests. Some rules are proposed for the application of MAUT.

Keywords: User's interests, MAUT, evaluation scheme

1 Using MAUT for Estimating Various Interests of a User

Estimation of a user's interests in recommender systems is very important, because these interests are the basis of the system's recommendations. There are many approaches for estimating the user's interest in the user modelling area. One approach is to ascribe the user *Multi-Attribute Utility Theory*, abbreviated as MAUT, (see [WE86]) as evaluation process. There are many approaches for estimating the user's interest and a great number of them do not explicitly mention using MAUT. Within this paper, we will have a detailed look at some of these approaches and examine the following issues:

- Can MAUT be used as common denominator of these approaches (although MAUT is not explicitly mentioned in the descriptions of these approaches)?
- Is it possible to define rules describing how to apply MAUT for estimating the user's interests?

For answering these questions, we sketch out MAUT in Section 2. The following Section 3 summarises a selection of recent approaches which estimate the user's interests and Section 4 proposes some rules for using MAUT.

2 Multi-Attribute Utility Theory (MAUT)

Multi-Attribute Utility Theory is an evaluation scheme which is very popular by consumer organisations for evaluating products. For example, in Germany "Stiftung Warentest" uses MAUT (see for example [Sti00]).[2]

According to MAUT, the overall evaluation *v(x)* of an object *x* is defined as a weighted addition[3] of its evaluation with respect to its relevant *value dimensions* [WE86]. The common denominator of all these dimensions is the *utility* for the evaluator. For example, a digital camera can be evaluated on the value dimensions *quality of image*, *flash*,

---

[1] This work has been supported by the EC through its IST-Programme under contract IST-1999-10688 CAWICOMS (see http://www.cawicoms.org for additional information).

[2] For supporting a person in evaluating an object by using MAUT, one could use sophisticated visualization tools, but these things are out of scope of this article.

[3] There are other possibilities for aggregation which are described by [WE86].

viewfinder, operation time, and handling (see [Sti00]). The overall evaluation is defined by the following *overall value function*:

$$v(x) = \sum_{i=1}^{n} w_i v_i(x)$$

Here, $v_i(x)$ is the evaluation of the object on the *i*-th value dimension d_i and w_i the weight determining the impact of the *i*-th value dimension on the overall evaluation (also called the *relative importance* of a dimension[4]), n is the number of different value dimensions, and $\sum_{i=1}^{n} w_i = 1$.

For each value dimension d_i the evaluation $v_i(x)$ is defined as the evaluation of the relevant attributes:[5]

$$v_i(x) = \sum_{a \in A_i} w_{ai} v_{ai}(l(a))$$

Here, A_i is the set of all attributes relevant for d_i, $v_{ai}(l(a))$ is the evaluation of the actual level *l(a)* of attribute a on d_i. w_{ai} is the weight determining the impact of the evaluation of attribute *a* on value dimension d_i. w_{ai} is also called *relative importance of attribute a* for d_i. For all d_i *(i=1,...,n)* holds $\sum_{a \in A_i} w_{ai} = 1$.

For example, for the *quality of images,* we have to consider attributes such as *sharpness, colour reproduction,* and *resolution,* whereas for *operation time*, we only have a single attribute, *actual operation time,* which reflects how long the camera can be operated.

In order to evaluate attributes, it is necessary to construct a scale representing the properties of the *levels* of an attribute. A scale from 0 (worst) to 10 (best) serves as measure of the evaluation.[6] Very often there is already a natural scale for the levels of the attributes. For example, for the resolution, we can define that 4 megapixel or more is best (10 points) whereas 0 megapixel is worst (0 points) (see Figure 1).

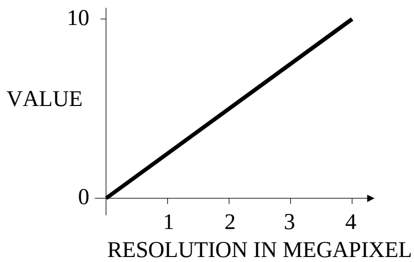

Figure 1: Example of a value function

Depending on the attribute, the evaluation function looks different. If it is a continuous variable, the evaluation function also will be a continues one, if it is a discrete variable, the function will be discrete. There is also the possibility to use value functions of

---

[4] The relative importance of a dimension also expresses the relevance of a dimension for the overall evaluation.

[5] There could be also *subdimensions* involved. In this case, the evaluation of the object on a dimension would be defined similar to the overall evaluation, by a weighted addition of the evaluation of the object with respect to its subdimensions.

[6] The boundaries of this scale can be selected to another values, e.g. [0..1], if one prefers another scale.

different shape, for example logarithmic ones, tri-partite intervals (not necessarily of equal length) or even user dependent scales. However, usually it is a good idea to use the simplest model possible for the domain for reasons of transparency (cf. Section 4.1.2).

Table 1 shows an example of the evaluations of the attributes of two different cameras which are relevant for the dimension *quality of images.* According to Table 1, camera CA is evaluated on *quality of images* as follows:

$$10 * 0,4 + 5 *0,3 +5 * 0,3 = 0,7$$

For camera CB, the evaluation is done accordingly. The evaluations of the cameras on their relative value dimensions are summarised in Table 2.

Attribute	**Evaluation of attribute of camera CA**	**Evaluation of attribute of camera CB**	**Relative weight on dimension *quality of images***
Sharpness	10	10	0,4
Colour reproduction	5	10	0,3
Resolution	5	10	0,3

Table 1: Evaluation of attributes and relative weights for dimension *quality of images*

	Evaluation of Products	
Dimension	CA	CB
Quality of image	7	10
Flash	10	1
Viewfinder	5	9
Operation time	10	2
Handling	9	3

Table 2: Examples for evaluations of digital cameras on their evaluation dimensions

Dimension	**Preferences (LP)**	**Preferences (SP)**
Quality of image	0,4	0,8
Flash	0,04	0,00
Viewfinder	0,1	0,1
Operation time	0,4	0,05
Handling	0,06	0,05

Table 3: Examples for user preferences

In Table 3 the preferences of two different persons are depicted. A studio photographer (SP) wants to make excellent pictures. There is no need for a long operation time, because recharging of batteries can be done easily and he never needs a flash. A landscape photographer (LP) who goes on long-term journeys (for example hiking in deserted areas) mainly wants to have a camera which has a long operation time since s/he will not be able to transport very many batteries. So, LP will prefer CA whereas SP will prefer CB (see Table 4).

	LP	**SP**
CA	8,24	7,05
CB	5,92	9,15

Table 4: Overall evaluations of cameras CA and CB for persons LP and SP

“Stiftung Warentest” uses MAUT over many years in order to evaluate products and presenting them to the public. Even newspapers print the results in a shortened form, by publishing the value dimensions, the evaluation of the products with respect to these dimensions and the overall evaluation of the products. This indicates that an evaluation according to MAUT can be easily understood . However, it should be noted that it is not trivial to identify the value dimensions of an object and of all attributes being relevant for the evaluation. Of course, it is much easier to understand them, once they have been defined. This is probably the reason why people use more simpler evaluation schemes when evaluating objects or just rely on the schemes of consumer organisations which they do understand.

In the following, we will examine some approaches for estimating the user’s interest and examine whether the used evaluation scheme will be compatible with MAUT.

3 Approaches for Estimating the User’s Interest

[EMM+01] present a system for news classification in the World Wide Web. The task of the system is to filter news for a single user. As seen from the view point of MAUT, the system has to determine the utility of a given article, i.e. whether it is interesting to the user or not. In the beginning, the user selects some *categories* from an ontology and assigns a *relevance value* to them. A news item is rated according to its relevance with respect to *categories*, *keywords*, and *resource channels*.

From a simplified viewpoint, the user has defined interest dimensions by specifying the categories which are of interest to him. An article is useful, if it is interesting which means that it falls in one of the selected categories. The value functions are equivalent with the function describing whether an article belongs to such a category or not. However, in this case it is not as simple as described in this text, because the offered categories do not match perfectly with the user’s interest dimensions: usually, they are too general. For this reason, keywords and resource channels are also used by the system of [EMM+01] for filtering the news articles. The weights which also determine how relevant, i.e. interesting, a news item is for the user, depend on (among other things) the relevance values assigned by the user.

[KOY01] presents a system which is able to learn the user’s preferences for scheduling meetings. The interests to be learned are variables such as duration, day-of-week, location, and start-time (regarding specific types of meeting). The system learns the user’s preferences through routine use which enables it to give customised scheduling advice. [KOY01] even assumes that the interests of the user slightly change over time.

From the viewpoint of MAUT, we have, in this case, a flat hierarchy. There are only attributes, but no intermediate value dimensions. For the attributes the value functions are different for each user. For example, the locations of the meetings regarding a specific type of meeting differ from user to user. Koychev’s system therefore tries to learn the value functions of the user. The weights of the evaluation function are not considered, because the system tries to predict the perfect combination

[LHL97] use a candidate critique model for interactive problem solving. In this approach, an automated problem solver presents candidate solutions to the user who gives feedback on these solutions. Based on this feedback, better solutions are searched and presented to the user.

The presented system is designed as constraint-based system and the use of MAUT is not obvious. However, for identifying the best solution in the user model preferences such as “prefers fewer stops” are represented which can be seen as value functions.

By using the candidate critique model, the system is not obliged to represent the user’s evaluation process, because it is the user who evaluates the objects. However, implicitly the system is learning the user’s value function, e.g. whether the user prefers more or less stops or which are the user’s most preferred airlines. In the same way, the weights are learned: if there is no solution which matches to all the user’s needs, the user has to decide whether to prefer a solution which fulfils his requirements regarding attribute A, but not B or the other way round. In the first case, the user expresses a greater weight for attribute A than B.

[CP01] explicitly uses MAUT for travel planning. So, it is clear that the approach is compatible with MAUT. Stereotypical information is used for not having to ask the user about all details.

4 Rules for Choosing the Right Complexity of MAUT

We have seen that all examined approaches in Section 3 ascribe an evaluation process to the user which is compatible with MAUT. It seems therefore not to be an issue whether to use MAUT or not, but to decide which complexity of MAUT to apply. There are many degrees at which complexity can be varied. In the following, rules are proposed for selecting a proper degree of complexity.

4.1 Rules

4.1.1 Hierarchy of dimensions and attributes

Description: In the simplest case, there are only attributes (cf. Section 2) and no higher level dimensions (such as the d_i in Section 2) which have been assigned weights. More complex models involve a hierarchy of dimensions and subdimensions.

Rule: There are two factors which determine the hierarchy:

- *The application domain*: If there are very few attributes or if the attributes cannot be subsumed in a hierarchy, then a non-hierarchical model has to be chosen.
- *Use of the model within the application*: If your domain allows to introduce a hierarchy to model the objects to be evaluated, it depends on the elicitation and presentation goals of your application what to use. If it is sufficient, to ask the user about the attributes and present them (in contrast to asking / presenting) higher level concepts, then a flat hierarchy may be chosen which just consists of attributes but not of dimensions.

4.1.2 Assumption of given weights and value functions

Description: Since it is possible that every user has its own value functions and weights, on the one hand, one approach is to try to estimate all these parameters for each user. On the other hand, one could assume some of these parameters as fixed and only estimate a part of them.

Rule: If there is no doubt about the parameter’s value, there is no use in trying to estimate them. However, in most cases it is not that simple, but there is a high number of a value functions which are agreed regarding their basic meaning. For example, nearly everybody agrees that a low price is better than a high price. In a simplification, the weights of attributes in the lower levels of a hierarchy also can be assumed as fixed. Which parameters of these should be assumed as fix depends on

- how much doubt about the actual value of this parameters (i.e. attribute weight and values functions) you have and
- whether it is possible to elicit the actual values of these parameters.

There may be a trade-off between these criteria. So, it can only be decided for each application separately. For example, even if there is some doubt about value functions and attribute weights, it may be reasonable to choose them as fix, because the domain is so complex that it is not feasible to ask the user about all of them.

4.1.3 Complexity of aggregation function

Description: Besides the weighted addition as aggregation function which were introduced here, there are more complex aggregation functions. It is possible to derive such function on its own, for example for introducing non-compensatory factors.

Rule: Keep it simple! The complexity of the aggregation function depends on the application domain, but it should be chosen as the simplest possible function.

4.2 Application example

Within the IST-project CAWICOMS[7] (see [CC00]) a configuration workbench is being developed for configuring complex products. The interface of this workbench has the task to support the user in choosing parameters which match the user's needs. Example domains are the configuration of complex telecommunication switches and IP-VPNs.

In order to help the user, the system proposes default parameter values and even sets parameter values for the user. For this purpose the user is ascribed MAUT as evaluation process (see [SM01]) The following design decisions have been taken:

- Hierarchy of dimensions: since the products to be configured are quite complex, the user cannot define every detail. For this purpose, the evaluation scheme consists not only of attributes but also of dimensions. In this way, the user can be asked about his/her interests regarding these dimensions. Based on this information, parameter values can be selected.
- Assumption of given weights and value functions: due to the high number of attributes, the user cannot be queried about each value function and each attribute weight. In addition, there is usually agreement on value functions and weights. So, they are assumed as fixed.
- Complexity of aggregation function: the use of a simple aggregation function (which uses the above described weighted addition) posed no problems. So, there was no need to use a more complex aggregation function.

---

[7] CAWICOMS is the acronym of "Customer-Adaptive Web Interface for the Configuration of Products and Services with Multiple Suppliers"

5 Conclusion and Future Work

We have examined some approaches for modeling the user's interests and therefore the user's evaluation function. It was shown that all these approaches are compatible with MAUT and differ in their complexity. Some rules have been proposed indicating when to use which complexity.

In this paper, only some work regarding the work of estimating the user's interests has been taken into account. It would be interesting to extend this survey including all relevant user modeling approaches for estimating the user's interests.

6 References

[CC00] CAWICOMS Consortium (2000). *Deliverable D01 – Requirements, Application Scenarios, Overall architecture, and Test Specification.*

[CP01] Chin, D., and Porage, A. (2001). *Acquiring User Preferences for Product Customization*. In Proceedings of the 8th International Conference, UM 2001.

[CP01] Chin, D., and Porage, A. (2001). *Acquiring User Preferences for Product Customization*. In Proceedings of the 8th International Conference, UM 2001.

[EMM+01] Eilert, S., Mentrup, A., Mueller, M. E., Rolf, R., Rollinger, C.-R., Sievertsen F., and Trenkamp, F. (2001). Bikini - user adaptive news classification in the world web web. In Schäfer, R., Müller, M. E., and Macskassy, S. A., editors, *Proceedings of the UM2001 Workshop on Machine Learning for User Modeling*, pages 37-47.

[Koy01] Koychev, I. (2001). Learning about user in the presence of hidden context. In Schäfer, R., Müller, M. E., and Macskassy, S. A., editors, *Proceedings of the UM2001 Workshop on Machine Learning for User Modeling*, pages 49-58.

[LHL97] Linden, G., Hanks, S., and Lesh, N. (1997). *Interactive Assessment of User Preference Models: The Automated Travel Assisstant*. In Proceedings of the 6th International Conference, UM97

[Sti00] Stiftung Warentest (2000). *Digitalkameras: Pixeljagd*. In: test 6/2000. Available via http://www.warentest.de/

[SM01] Schütz, W. and Meyer, M. (2001). Definition einer Parameter-Hierarchie zur Adaptierung der Benutzer-Interaktion in E-Commerce-Systemen. In *Proceedings of ABIS 2001*.

[WE86] Winterfeld, D. von and Edwards, W. (1986). *Decision Analysis and Behavioral Research*. Cambridge, England: Cambridge UniversityPress.