User-Driven Ranking for Measuring the Interestingness of Knowledge Patterns

M. Baumgarten
Faculty of Informatics,
University of Ulster,
Newtownabbey, BT37 0QB, UK

A.G. Büchner
Faculty of Informatics,
University of Ulster,
Newtownabbey, BT37 0QB, UK

J.G. Hughes
University of Ulster,
Coleraine, BT52 1SA, UK

Choosing thresholds for knowledge discovery algorithms to achieve a set of results which either solve a specific problem or are optimised towards the users desires requires the re-run of the algorithm, which is a tedious and time-consuming procedure. Within this paper an importance based interestingness measure is introduced that can be applied as a posterior phase. In addition three manipulative operations (balancing, boosting and inversion) are outlined operating upon results of data mining exercises to allow the interpretation of the same result from different viewpoints. The overall framework represents a novel user-driven ranking procedure, allowing measuring the interestingness of knowledge patterns utilising a combination of qualitative and quantitative indicators as well as a user driven importance or interestingness value.

Keywords: data mining, interestingness, knowledge manipulation, knowledge interpretation

1 Introduction

The discovery of knowledge patterns has been praised as the panacea for finding hidden and valuable information in large amounts of data. However, the majority of such data mining exercises produces huge result sets, which are either incomprehensibly large or meaningless in a given problem space. Furthermore, different domains require different viewpoints on the same result, which, apart from some visualisation techniques, are not available to date and require a re-run of the selected data mining algorithm utilising different parameters. For instance, in order to provide useful knowledge for a marketer (interested in click-to-close ratios on a web site) and a web administrator (interested in effective load balancing of the site), two separate sets of knowledge patterns have to be derived.

The main emphasis of knowledge discovery has been twofold. The first has concentrated on the development of the most optimal creation of results with respect to memory usage and speed. The second has been concerned with various pre- and post-processing mechanisms to effectively discover patterns and to decide if the discovered patterns should be part of the result set or not, based on given constraints.

While such constraints minimize the search space and thus optimise computation, all approaches are focusing on quantitative measures such as number of items, support and confidence. Or, to be more candid, once again computational optimisation has been given priority over user (customer) requirements. The objective of this paper is to overcome this bias and to introduce the concept of user-driven ranking for measuring the interestingness of knowledge patterns, and thus combining qualitative and quantitative measures.

Qualitative measures, which, to date, have not yet been considered, in much detail, by the data mining research community, offer unique novel features. Firstly, they allow domain-specific ranking of results as opposed to traditional domain-agnostic ordering. Secondly, they are mainly algorithm independent, which allows their integration into existing environments. Thirdly, the created result sets can be manipulated further in ways unbeknownst to the type of patterns, which will be demonstrated by newly introduced balancing, boosting and inversion operations. Lastly, qualitative measures can be combined with existing quantitative measures.

The paper is organized as follows. In Section 2 related work is recapitulated and drawbacks are shown. In Section 3, a ranking framework is built and notational issues are addressed. Section 4 provides a novel approach to rank patterns based on weights provided by users as well as discovered constraints. In Section 5 manipulative operations on distributions are presented, which provide a facility to interactively work with result sets. Section 6 outlines architectural and implementation issues and shows a potential application in the web mining arena. Section 7 concludes the paper and outlines future work.

2 Related Work

The general objective of any pattern detection method is that of discovering the patterns itself and to allocate a number of describing information to it like the number of occurrences for repeating patterns, probability measures or structural information. However, this is not always an easy task, since the discovery of knowledge patterns from large amounts of data is computationally expensive and inherently hard [5]. This is caused by an explosion of possible permutations and combinations in a given data set.

Due to the high volume of patterns discovered by pattern detection algorithms there is also a lack of usability of the resulting patterns. A number of researchers have addressed this problem and developed various methods to measure the interestingness of a pattern.

Thresholds, in form of numeric minima and maxima, which bias the search and the result space, have been provided by almost all detection mechanisms [3]. The most relevant for associative and sequential patterns are support (number of records in the data that satisfy the pattern) and confidence (belief that if the antecedent of a pattern is true that also the consequent is true). A further threshold of patterns could be related to its structure such as minimum and maximum number of items within a pattern.

It has been the objective by more recent endeavours to bias the search through the usage of additional constraints and advanced domain knowledge. The incorporation of such user-driven domain knowledge not only reduces the result space, it usually also minimizes the search space and provides patters of higher generality.

One such approach that has been proposed allows the specification of regular expression-based pattern templates in the form of SQL-like queries [7] as well as concept hierarchies that can be provided prior to the discovery [2].

[6] tackles the central problem of good measures to identify the interestingness of a pattern. Two different kinds of interestingness measurements were introduced. Objective measurements related to the structure of a pattern object and the underlying data used to discover them, while subjective measurements depend on the user's needs, the domain the data is analysed in, and the scenario they are applied to.

[8] have introduced the concept of weighted association rules, which provide some ideas for our approach. Their work allows the discovery of associations, which have been allotted additional information that represent the occurrence of an item in a transaction. Furthermore, weighted association rules use metric density measures to skew the result set by finding ranges of weights for certain items.

Another approach that has motivated the introduction of a qualitative measurement of sequential patterns is that of costing which has been embedded in the classification algorithm C5.0 [4]. Basically, classification labels can be allotted differential penalty costs for misclassification, which are then considered during the discovery process.

However, the majority of presented research concentrates on minimizing the result space. They neither provide any type of importance or relationship among discovered patterns which can be used to rank them, nor do they allow the interaction with the result set. Both issues will be addressed in the reminder of this paper.

3 Ranking Framework

In order to propose the outlined user-driven ranking of knowledge patterns across multiple interestingness measurements as well as manipulation operations upon ranked results, some basic constructs are required, which are introduced in this section.

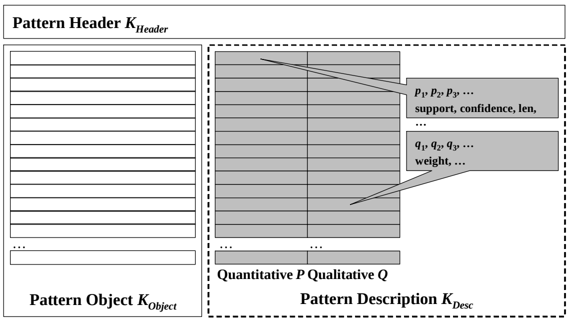
Pattern Header K_{Header}
$p_1, p_2, p_3, \ldots$
support, confidence, len, …
$q_1, q_2, q_3, \ldots$
weight, …
…
… …
Quantitative P Qualitative Q
Pattern Object K_{Object}
Pattern Description K_{Desc}

Figure 1. Knowledge Pattern K

A general ranking framework is presented, which is designed to work on the result set of a pattern discovery exercise, independent from its type or implementation and thus algorithm-specific implementations or adaptations are not necessary. However, patterns detection methods have to provide the dimensions selected by the user to provide a ranking across multiple dimensions.

K_{Header} provides meta information about the result (for instance, thresholds chosen for the algorithm or the number of objects in the result set). K_{Object} contains the actual patterns, which are determined by the type of data mining algorithm that is used. Both elements of a knowledge pattern are ignored in the rest of this paper, and it is solely concentrated on the descriptive part. K_{Desc}, which provides numeric information about each pattern object, consists of the tuple $K_{Desc} = <P, Q>$, where $P = \{p_1, p_2, p_3, \ldots\}$ represents quantitative measures and $Q = \{q_1, q_2, q_3, \ldots\}$ represents qualitative measures.

Once each quantitative or qualitative measure is normalised (see Section 4), it is represented as a distribution $\Omega = \{\omega_1, \omega_2, \omega_3, \ldots\}$. If an algorithm provides values that are already normalised, there is no requirement to normalise them again; in fact it would potentially be counterproductive. Thus, normalisation only applies to non-normalised values either provided by a data mining algorithm or through derivation.

Based on these constructs an example data set is shown in Table 1 containing K_{Desc} for 20 patterns, which will be used throughout the paper to demonstrate the proposed methods. Table 1 contains three quantitative and a single qualitative measure that are common in sequential, associative and episodic patterns. Other data mining techniques provide different types of measures which can be handled by the proposed framework. Conforming to the notation outlined above, the example can be expressed as K_{Desc} = < {length, support, confidence}, {weight}>.

ID	Length	Support	Confidence	Weight
1	9	41	35	0.77
2	9	55	36	0.46
3	2	90	48	0.51
4	1	74	43	0.59
5	4	1	34	0.87
6	6	75	87	0.70
7	9	2	92	0.41
8	7	68	18	0.62
9	2	20	5	0.42
10	10	62	93	0.20
11	3	95	98	0.40
12	10	8	98	0.20
13	3	3	13	0.95
14	8	55	28	0.30
15	5	41	22	0.99
16	6	80	17	0.29
17	9	46	63	0.86
18	2	12	27	0.48
19	6	95	86	0.79
20	3	51	84	0.25

Table 1: Example K_{Desc}

The example contains four measurements. It has to be stressed that K_{Desc} is not limited to these measures; it can be extended to contain any number of quantitative or qualitative measures as long as all measures comply with the definitions outlined within Section 4.1 and 4.2.

4 Quantity and Quality Measures

Knowledge patterns provide a number of qualitative and quantitative measurements and each element in K_{Desc} provides a basic importance relationship by itself, which can be used by ordering the result set in ascending or descending order. For instance, when measuring the click-to-close rate on a web site, short patterns are desirable, whereas long patterns are beneficial when looking for high volume purchases in an e-commerce scenario. Accordingly, support and confidence of a pattern can have impact in other scenarios. For instance, a high support value could refer to a high buyer's scenario and also to a high fraud rate; a high confidence level indicates a high browser-buyer conversion rate, but also a high browser-abandonment ratio. These operations are limited, since they relate only to one quantitative measure.

Qualitative measures are derived from the pattern object itself or more precisely from its structure and / or its content utilising additional domain knowledge like weights, temporal or structural information. While it is possible to incorporate such measures exclusively in the post-processing stage, with more complex structures and domain specific constraints, this could be difficult and incoherent. Additionally, by incorporating them in the pre-processing or discovery stage, these measures can further be used to constrain the search space by specifying minima and maxima.

A more effective way to measure the importance of a pattern is to use qualitative measures. An example is the overall pattern weight derived through weighing each of its items. Ordering by this qualitative measurement provides a more accurate ranking of the result set since a pattern containing items of high priority gain a higher importance value. However, this method is based on a single measurement discarding all other measures. To overcome this limitation a ranking value is introduced in Section 4.3, Measurement Coalescence, combining qualitative and quantitative measures described in Sections 4.1 and 4.2. This measurement coalescence is based on a relevance threshold providing an accurate ranking among patterns tailored towards the user's interestingness and domain or scenario.

4.1 Qualitative Measurement

Given a set of weighted items $I = \{i_1, i_2, i_3, \ldots\}$, where each i is represented as a pair $<v, w>$, v denoting the item value and w the allotted weight, $(0 \leq w \leq 1)$. Table 2 shows an artificial example of such a set of weighted items.

There are different ways to allocate weights to items, for instance based on background knowledge or statistical methods such as confidence, or in a user-defined manner. The pattern weight pw is calculated as following:

$$pw = \frac{\sum_{1}^{|i|} w(i)}{n} \qquad \text{Eq (1)}$$

Other methods may be used to calculate an overall weight of a pattern taking into account a more complex pattern object structure or specialized weight distributions. However it is compulsory that $0 \leq pw \leq 1$.

Item value	Weight
Purchase.html	1.00
Evaluation.html	0.80
Pricing.html	0.65
Download.htm	0.65
…	…
Support.html	0.40
WhitePapers.html	0.35
Company.html	0.25
Help.html	0.10
Disclaimer.html	0.10

Table 2: Example Weighted Item Values

4.2 Quantitative Measurement

Quantitative information provides structural or probability related knowledge about the result patterns. Examples for such knowledge are the number of items in a pattern, support, confidence, etc. They are usually derived from the discovery process and connected to a minimum / maximum threshold to constrain the search and result space. Similar to qualitative measures, each of these measures themselves reflect a certain importance level, which can be used by ordering the result set by this measure. However, analysing the importance of a pattern by applying only a single measurement lacks the importance of all other dimensions. Combining information by normalizing their range to the interval [0…1] and connecting it to a user-defined relevance value (see Section 4.3) provides a user-driven quantitative importance measurement that relies on all available or selected dimensions.

The normalization of quantitative values can be formalised as following.

$$w = \frac{l - l_{\min}}{l_{\max} - l_{\min}} \qquad \text{Eq (2)}$$

Using the normalization equation shown in Equation 2 normalises a numeric value to the range [0…1], where $(l / l_{max}) * l_{min}$ and l_{max} $[l / l_{max} \ldots 1]$ describe the borders respectively. Assuming that ω is an importance measure this would lack in the importance of any value equal to l_{min}., since these values would be normalized to 0. Thus l_{min} has to be set to 0 to guarantee that $\forall \omega_i \in \omega : \omega_i > 0$. Henceforth, the equation above can be simplified as following.

$$w = \frac{l}{l_{\max}} \qquad \text{Eq (3)}$$

By applying the simplified normalisation form to the qualitative pattern description Q a normalized co-efficient ω is derived for each q_k, which will be used for further processing.

4.3 Measurement Coalescence

Within this section a ranking value α is introduced combining all normalised qualitative and quantitative values in K_{Desc}. Additionally, an importance factor β is introduced to provide a user-driven relevance value for each element in K_{Desc}. Again, when analysing the click-to-close ratio, the length of a pattern is more relevant than the confidence factor. The calculation shown in Equation 4 combines qualitative and quantitative measures into a single value within the range [0…1], where each element in K_{Desc} is of equal importance.

$$a = \frac{\sum_{1}^{|w_p|} (w_p) + \sum_{1}^{|w_q|} (w_q)}{|K_{Desc}|} \qquad \text{Eq (4)}$$

Although the above approach provides a single ranking measurement covering multiple dimensions, it does not provide a user-driven ranking of the result set since all values in K_{Desc} are treated as equally relevant. To overcome this shortcoming, the importance factor β is incorporated for each element in K_{Desc}, such that that $0 \leq \beta \leq 1$. Equation 5 shows the incorporation of, β based on Equation 4

$$a = \frac{\sum_{m=1}^{|b_p|} (b_{p_m} * w_{p_m}) + \sum_{n=1}^{|b_q|} (b_{q_n} * w_{q_n})}{\sum_{1}^{|b_p|} b_p + \sum_{1}^{|b_q|} b_q} \qquad \text{Eq (5)}$$

4.4 Example Rankings

Applying the methods outlined above to the example K_{Desc} shown in Table 1, the ω distributions shown in Table 3 have been derived.

Combining all ω distributions into a single ranking value α results in an overall ranking as shown in Figure 2.

The “Original” distribution based on Equation 4 does not consider any importance measures. The distributions “Ranked 1” and “Ranked 2” are calculated using Equation 5 highlighting the ranking change for different $b_{w_w}, b_{w_l}, b_{w_S}, b_{w_C}$. The settings for “Ranked 1” are β_π = 10, β_l, = 100 β_S = 100, β_C = 100. The settings for “Ranked 2“ are β_π = 80, β_l, = 100 β_S = 30, β_C = 10.

ID	ω_I	ω_S	ω_C	ω_W
1	0.9	0.41	0.35	0.77
2	0.9	0.55	0.36	0.46
3	0.2	0.90	0.48	0.51
4	0.1	0.74	0.43	0.59
5	0.4	0.01	0.34	0.87
6	0.6	0.75	0.87	0.70
7	0.9	0.02	0.92	0.41
8	0.7	0.68	0.18	0.62
9	0.2	0.20	0.05	0.42
10	1.0	0.62	0.93	0.20
11	0.3	0.95	0.98	0.40
12	1.0	0.08	0.98	0.20
13	0.3	0.03	0.13	0.95
14	0.8	0.55	0.28	0.30
15	0.5	0.41	0.22	0.99
16	0.6	0.80	0.17	0.29
17	0.9	0.46	0.63	0.86
18	0.2	0.12	0.27	0.48
19	0.6	0.95	0.86	0.79
20	0.3	0.51	0.84	0.25

Table 3 Example ω distributions

The behaviour can be demonstrated by the two highlighted areas in Figure 2. In area *a,* α is lowered for "Ranked 1" and raised for "Ranking 2", while for area *b,* α is raised for "Ranked 1" and lowered for "Ranking 2". All series' will be identical for $b_{w_w} = b_{w_I} = b_{w_S} = b_{w_C} > 0$.

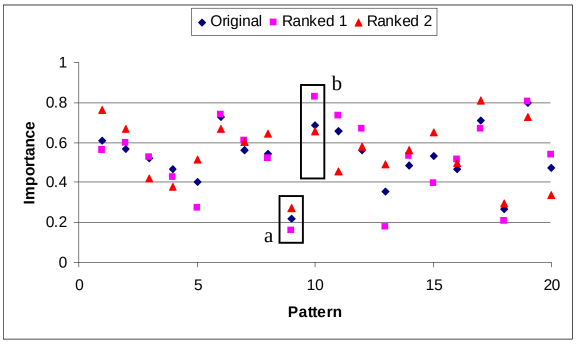

Figure 2: Ranking

5 Manipulating ω Distributions

In order to enable modifications to an entire ω distribution a host of manipulation methods is introduced, namely *balancing*, *boosting* and *inversion*.

The objective of such modifications lies in the requirement of adapting the importance measure reflected by ω to different scenarios. For instance, in the click-to-close example in the previous section, the objective was to decide which elements in K_{Desc} are of greater importance based on a given scenario. Changing the focus to the actual values or the distribution thereof has required a re-run of the algorithm so far. Applying different manipulation methods allows the interpretation without this additional time-consuming step.

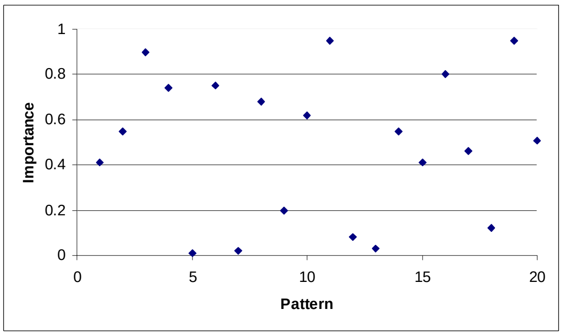

Figure 3: Example ω Distribution

Manipulation methods allow to boost, balance or invert a given ω distribution. Figure 3 shows the example ω_S based on Table 3 that has been used to visualize the impact of different manipulation operations.

5.1 Boosting and Balancing

Boosting allows a raise / lowering of a ω distribution, based on a user-defined threshold $0 \leq k_B \leq 1$, where $k_B = 0.5$ reflects the original distribution. In Figure 4 the example distribution shown in Figure 3 is raised to 80% and lowered to by 20%, respectively. Applying this method also condenses the distribution to the lowest or highest level, respectively.

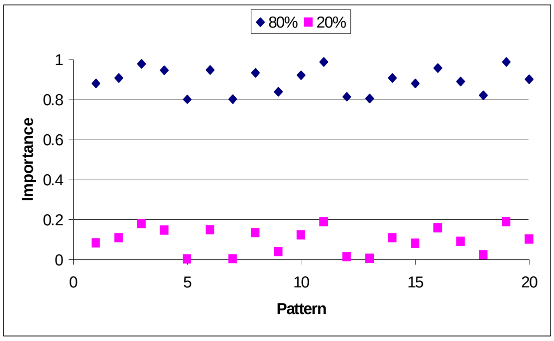

Figure 4: Boosting (80%, 20%)

Balancing arranges the distribution around a certain split threshold s ($0 \leq s \leq 1$). The user-defined threshold $-1 \leq k_L \leq 1$ dictates whether values are raised or lowered. If $k_L < 0$ then all values greater than *s* are raised, whereas all values less than *s* are lowered towards the boundaries of the range, respectively (also known as stretching). If $k_L > 0$ then all values greater than *s* are lowered, whereas all values greater than *s* are lowered towards the threshold *s*, respectively (squeezing).

In the example below, the split threshold *s* is set to 0.5, which is defined as the centre of a normalised range. Increasing k_L results in a lowering of values above *s* and a raise of values below *s*. Figure 5 (80%)

shows an example where the original distribution shown in Figure 3 is arranged around *s*. The -80% example demonstrates a widening towards 0 and 1, respectively.

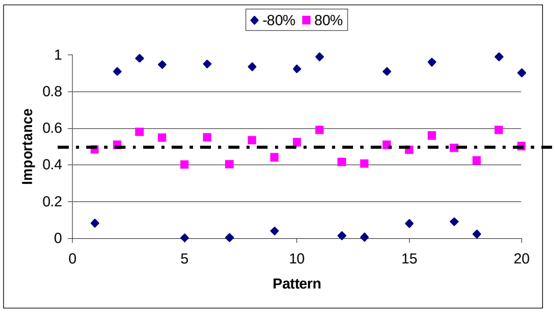

Figure 5: Balancing (-80%, 80%)

5.2 Inversion

Inversion is an important manipulation method, which allows to reverse the importance of a given ω distribution. Increasing a user-defined threshold $0 \leq k_I \leq 1$ condenses a given ω distribution towards 0.5, which is defined as the centre of the normalised range, so that all values equal to 0.5 for $k_I = 0.5$. If k_I is increased further it widens the distribution in inverse order so that each $\omega_i \in \omega$ is equal to $1 - \omega_i$ for $k_I = 1$. Figure 6 shows the change in distribution for $k_I = 0.45$ and $k_I = 1$, respectively.

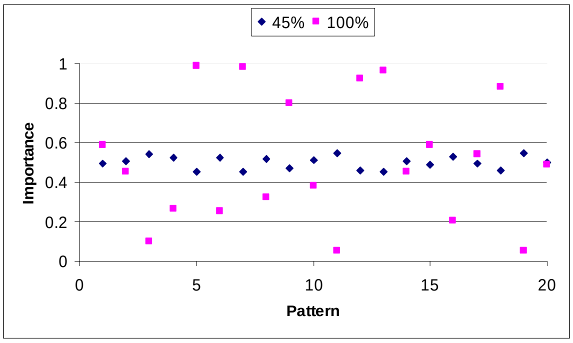

Figure 6: Inversion 45%, 100%

5.3 Concatenation of Manipulations

The introduced manipulations maintain the format of ω distributions, which means that the output of a manipulation can be used as input for another manipulation.

For instance, an already balanced distribution can be boosted towards a certain level. This allows to split a given ω distribution at any level (balancing) and position it anywhere within the given range (boosting). An example is shown in Figure 7.

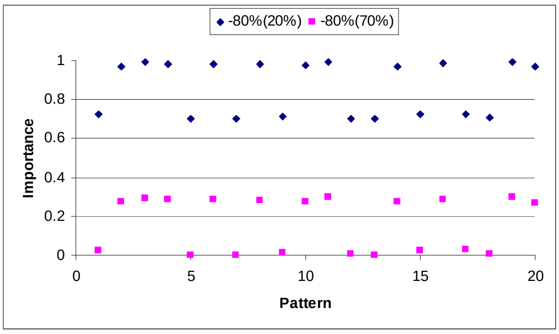

Figure 7: Balancing à Boosting
(-80%/20%, -80%/70%)

6 Implementation and Application

The *implementation* of the proposed ranking framework to extend traditional pattern discovery applications is straightforward since it does not affect existing knowledge pattern environments. The proposed methods are designed to be applied to the result set itself and are agnostic to the process of pre-processing, discovery, and post-processing. This, on the other hand, requires that a given pattern detection algorithm provides all qualitative and quantitative measures that a user requires for the usage with the proposed ranking approach. While some of these measures can be derived from the result set, such as the size of a pattern, is could potentially lead to inconsistencies, given that some algorithms define certain measures in different ways than others.

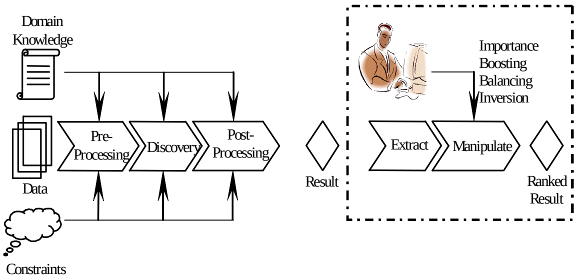

Figure 8: Implementation Architecture

For ranking purposes the way a measure is derived is irrelevant, as long as all the measures follow the definition as outlined in previous sections. Figure 8 visualizes the general implementation architecture.

This architecture allows the separation of the proposed approach from any discovery process and the connection to any advanced ranking tools. Such a tool could be provided in the form of a result browser through enabling a user to view the same result set from the scenario's viewpoints, without the need to re-execute the time-expensive task of finding all desired knowledge patterns.

To validate the proposed methods an example *application* has been selected within the area of web mining and a small sample has been extracted from a result set. The sample, shown in Table 4 shows four sequences representing online behaviour.

K_{Desc}					K_{Object}
ID	ω_l	ω_S	ω_C	ω_w	
1	0.5	0.9	0.2	0.5	index.html, purchase.asp
2	0.5	0.4	0.7	0.9	banner.html, purchase.asp
3	0.75	0.2	0.4	0.7	index.html, banner.html, purchase.asp
4	1	0.1	0.05	0.2	index.html, banner.html, help.html, contact.html

Table 4: Sample Set

Traditionally, ordering by any single value of K_{Desc} is supported, providing an importance among all sequences. However, combining all measures into a single value proves beneficial because one can order the result set over multiple dimensions. Table 5 shows the ranking value for 6 different scenarios, where $\alpha_0 = (\beta_\pi = 0.5, \beta_S = 0.5, \beta_C = 0.5, \beta_l = 0.5)$ represents an equal importance of all measures in K_{Desc}, ranking the pattern in the following order 2, 1, 3, 4.

ID	α_0	α_1	α_2	α_3	α_4
1	53	55	52	58	64
2	63	55	56	54	50
3	51	30	31	29	63
4	34	8	7	8	73

Table 5: Importance based Ranking

When analysing the buying pattern (indicated by purchase.asp) a combined measurement of support and confidence is naturally more useful. Eliminating weight ($\beta_w = 0$) and the number of items ($\beta_l = 0$) provides such a measurement $\alpha_1 = (\beta_S = 0.5, \beta_C = 0.5)$. If the conversion rate is more important than the number of purchases, $\alpha_2 = (\beta_S = 0.5, \beta_C = 0.6)$ is a possible setting. Analogously, if the number of customers who purchased a product is of higher relevance than the conversion rate $\alpha_3 = (\beta_S = 0.6, \beta_C = 0.5)$ can been chosen. Table 5 shows the ranking order for each setting.

When identifying visitors who get lost in a web site, long patterns with low confidence are important. In order to model low confidence in the result set, the inversion operation is applied ($K_I = 100$ for ω_C). A sensible setting for the distribution is $\alpha_4 = (\beta_w = 0.5, \beta_S = 0.3, \beta_C = 0.8, \beta_l = 1)$.

Although these examples are limited due to the short number of patterns in the result set, they show the potential of the proposed framework.

7 Further Work and Conclusions

Within this paper a novel approach has been proposed to provide a user-driven ranking measurement based on multiple qualitative and quantitative measures. In addition several manipulation and importance techniques are discussed to enable the requirements of different scenarios and specific domain interests. Although, none of the proposed methods optimises the discovery process itself, it does have implications on the quality and usability of discovered pattern.

Weighted items have been introduced which have proven to be highly beneficial in certain scenarios. Taking this approach a step further, it is possible to put a weight on item sets as opposed to items per se. In fact, item sets would then be handled as a special case of items. This approach provides even more powerful representations of distributions, for example the allotment of weights to certain genetic sequences, which are known to cause certain diseases. However, the introduction of a higher granularity of weights, presents new challenges, such as the resolution of ambiguities in overlapping item sets.

References

[1] R. Agrawal, R. Srikant. Mining Sequential Patterns, Proc. 11th Int'l Conference on Data Engineering, pp. 3-14, 1995.

[2] M. Baumgarten, A.G. Büchner, S.S. Anand, M.D. Mulvenna, J.G. Hughes. Navigation Pattern Discovery from Internet Data; B. Masand, M. Spiliopoulou (eds.) Advances in Web Usage Analysis and User Profiling, Springer-Verlag, pp 74 – 91, 2000

[3] W. Klösgen, J. Żytkow (eds.). Handbook of Data Mining, Oxford University Press, 2002.

[4] R. Quinlan, C5.0: www.rulequest.com, 1998.

[5] G. Piatetsky-Shapiro: Knowledge Discovery in Databases, AAI/MIT Press, 1991

[6] Silberschatz, A. and A. Tuzhilin, "What Makes Patterns Interesting in Knowledge Discovery Systems," IEEE Transactions on Knowledge and Data Engineering 8, pp 970-974, 1996.

[7] Myra Spiliopoulou and L.C. Faulstich. WUM: A Tool for Web Utilization Analysis. In *EDBT Workshop WebDB'98*, Valencia, Mar. 1998. Springer Verlag. Ext. version in LNCS 1590.

[8] W. Wang, J. Yang, P. Yu.; Efficient Mining Weighted Association Rules (WAR), *Proc. of the 6th ACM SIGKDD Int'l. Conf. on Knowledge Discovery and Data Mining*, pp. 270-274, 2000.