A Taxonomy of Attribute Scoring Functions

J. Schmid and J. Bernard

1 University of Zurich, Switzerland

Abstract

Shifting the analysis from items to the granularity of attributes is a promising approach to address complex decision-making problems. In this work, we study attribute scoring functions (ASFs), which transform values from data attributes to numerical scores. As the output of ASFs for different attributes is always comparable and scores carry user preferences, ASFs are particularly useful for analysis goals such as multi-attribute ranking, multi-criteria optimization, or similarity modeling. However, non-programmers cannot yet fully leverage their individual preferences on attribute values, as visual analytics (VA) support for the creation of ASFs is still in its infancy, and guidelines for the creation of ASFs are missing almost entirely. We present a taxonomy of eight types of ASFs and an overview of tools for the creation of ASFs as a result of an extensive literature review. Both the taxonomy and the tools overview have descriptive power, as they represent and combine non-visual math and statistics perspectives with the VA perspective. We underpin the usefulness of VA support for broader user groups in real-world cases for all eight types of ASFs, unveil missing VA support for the ASF creation, and discuss the integration of ASF in VA workflows.

1. Introduction

We refer to attribute scoring functions (ASFs) as data transformations from an input data attribute to a numerical distribution of output scores. ASFs naturally include concepts from statistical normalizations, attribute numerification [JFJJ08], and transfer functions [LKG+16]. ASFs are special in that output scores carry valence information: high output scores necessarily imply goodness, whereas low scores indicate neutral or even negative sentiments. Finally, output scores of ASFs have a polarity characteristic, which may either be unipolar (e.g., [0..1]) or bipolar (e.g., [−1..1]). The semantics of both valence and polarity allows users to express their preferences for attribute values through the creation of an ASF. The functional characteristics of ASFs can always be specified in a non-visual way by users with a programming background (the MATH perspective). As an alternative, many ASFs can already be created with interactive visual interfaces, allowing the ASF creation and adjustment by larger user groups (the VA perspective). With multiple ASFs at hand, the output of multiple ASFs is often (interactively) weighted [WDC+18] and combined [GLG+13] to facilitate multi-criteria decision making at the granularity of items. One of many analysis scenarios is finding a used car (the item-level) by defining preferences for attributes of interest. Preferably, a car may, e.g., be a) as cheap as possible, b) as fast as possible, c) neither too old nor too new, d) black, silver, or at least blue, and e) particularly have as few exhausts as possible (with high emission being penalized exponentially).

The analysis goal in this scenario is multi-attribute ranking, an approach to structure potentially large and complex item collec-
tions by computing ranks based on attribute scorings. An example VA approach with ASF creation support is LineUp [GLG’13] that offers an interactive user interface, as well as a scripting interface to enable the creation of ASFs for different user groups. Another analysis goal is multi-criteria optimization [PB93], where the challenge is to select compromise solutions using optimization algorithms. A VA example with a connection to ASFs is PAVED [CMMK20], with an emphasis on Pareto front visualizations that support users in applying their preferences for selecting the most preferred solutions. A third goal related to ASFs is similarity modeling [SJ99] in information retrieval where distance metrics are applied either on individual attributes [BRS’17] or the entire attribute set [Pand06]. The results of distance metrics form 1D metric spaces and serve as the input for ASFs, which are then inverted (from distances to similarities) and finally non-linearly scaled [MZ11] to better approximate the notion of similarity as perceived by humans [She87].

Based on the reflection on literature in MATH and pioneer approaches in VA, the motivation for this research was manifold. First, it may make sense to distinguish between the study of ASFs and tools enabling users to create different types of ASF. Second, we identified that both ASFs and ASF creation tools have not yet been investigated and described systematically. Third, research on the creation of ASFs in MATH and VA seems to co-exist independently from each other, leaving room for the study of combinations and synergies. Fourth, surprisingly little support exists for the interactive creation of ASFs; for some types of ASFs VA support is still entirely missing. Finally, in several real-world use cases, we identified the unsatisfied need of non-experts for the interactive creation of ASFs, which additionally calls for future VA support.

Our primary contribution is a hierarchical taxonomy for ASFs directly drawn from patterns observed in both the MATH and the VA literature. The main distinction criterion is between the categorical or numerical input attributes, further sub-divided by the structural characteristics in both branches. Our secondary contribution is an overview of tools enabling users to create ASFs. In this overview, we utilize the descriptive power [BLD04] of the taxonomy to categorize and unify tools from MATH and VA. We demonstrate the usefulness of both the taxonomy and the tool overview for all eight ASF types with a) examples of real-world ASF instances, b) references to (VA) tools enabling the ASF creation, and c) descriptions of areas in the design space not yet supported with VA. Finally, we discuss the integration of ASFs in VA workflows for different analysis goals and outline future work. This work is meant to enable a deeper understanding of different types of ASFs and their interactive creation with VA tools for experts and non-experts.

2. Attribute Scoring Functions
2.1. Characterizing Attribute Scoring Functions

We characterize ASFs according to the three aspects data transformation, polarity, and valence, allowing to transform attribute values according to user-defined preferences.

Data Transformation: ASFs are data transformations that assign scores to attribute values. The mapping of ASFs is always defined for the entire input value domain. By design ASFs can be described formally, to enable automatic execution. In contrast to most machine learning models trained in a learning process, the functional behavior of ASFs is described in a creation step before execution.

Polarity: The value domain of output scores can be unipolar (from neutral to one extreme value) or bipolar (from a negative extreme value to a positive extreme value). A unipolar interval [0..1] [GLG’13] may be a good default solution for ASFs, just like the min-max normalization. However, analysis scenarios exist where a bipolar interval [−1..1] is more useful: bipolar scores allow modeling criteria for both, particularly good and bad items.

Valence: By default, high score values indicate something positive (high is good). However, in some cases, it is useful to enable users to flip the valence of the ASF to determine that low output values should yield the highest scores (low is good), e.g., for a price attribute.

2.2. Creation of Attribute Scoring Functions

Every ASF can be created by programmers using MATH support. As an alternative to MATH, some ASFs can also be created with VA support, e.g., to open the process for larger user groups. Inspired by a VA pioneer approach [GLG’13], a conceptual interface for the ASF creation is shown in Figure 1 (right), using a non-linear two-point ASF as an example. We illustrate how users can create and modify ASFs interactively; here, by dragging a selected orange line segment in 2D, leading to adoptions of the functional behavior. We show the distributions of both input values (blue, from the bottom) and output scores (blue, to the right), as it may be recommendable to enable users to observe the effect of the data transformation.

2.3. Attribute Scoring Functions in the Literature

We conducted an extensive cross-domain literature research that aimed at including both the MATH and the VA perspective. Our search target was two-fold as we sought two different classes of research items: 1) concrete ASFs and 2) ASF creation tools. Both sets of retrieved items helped us to analyze commonalities and differences across ASFs and to identify their structural characteristics. One result of this iterative process is the taxonomy of ASFs presented in Figure 1 (left), described in Section 3. The taxonomy also supported the creation of the structured overview of tools enabling users to create ASFs, presented in Table 1, described in Section 4.

3. A Taxonomy of Attribute Scoring Functions

The literature research revealed a space of possible types of ASFs. Design targets when identifying and structuring types of ASFs in this space were a) to differentiate between structural characteristics of ASFs and b) to assess the complexity of ASFs when being created with VA support. As MATH may become arbitrarily complex, we decided to only present types of ASFs for which plausible real-world examples exist, to further motivate the creation of ASFs in practice. The result of this process is a hierarchical taxonomy of ASFs, presented in Figure 1 (left). In the taxonomy, the most principal distinction is between ASFs for categorical and numerical input attributes (Sections 3.1 and 3.2), as for both MATH and VA the creation of ASFs fundamentally differs for these two attribute types. In this section, we provide an overview of the taxonomy and describe types of ASFs in detail. We explicitly point towards spaces for ASF creation that do not contain VA solutions yet.

© 2021 The Author(s)
Eurographics Proceedings © 2021 The Eurographics Association
3.1. Attribute Scoring Functions for Categorical Attributes

We distinguish between Score Assignment and Ordered variants for the transformation of categorical attributes with a fixed number of categories (values, observations, levels) into numerical scores. Score assignment uses a lookup table with preferences to assign categories to scores. Ordered ASFs require an ordering of categories first, before allocations of output scores can be achieved.

**Score Assignment**: A straightforward ASF for categorical values is a lookup table with a number for each category. In the illustration, three high and two comparatively low scores have been defined for the five categories. In a house purchasing example, Score Assignment may be used to assign preferences for districts or architecture types.

**Ordered, Equidistant**: Equidistantly ordered ASFs transform ordered categories to uniform distributions across the output space. This order of categories can be the result of a distance function for categories [BRS12] or be based on interactive orderings performed by users, e.g., via drag-and-drop as presented in Podium [WDC18] for item ranking. For a shopping dataset with shoes, this would be beneficial if a user is sure about the ordering of color preferences, but cannot quantify this color comparison.

**Ordered, Non-Equidistant**: Non-equidistant ASFs extend the functionality of ordered equidistant ASF by one degree of freedom: the transformations of ordered categories into scores do not need to follow a uniform distribution. This supports skewness tweaks, which can be quite beneficial for semantically unbalanced categories. Johansson et al. employ a non-equidistant ASF where differences in scores are the results of user-defined similarities across categories [JFJJ08]. For a traveling example, a user may prefer very few countries over many uninteresting ones. Visual interfaces may enable users to allocate large parts of the score value domain only for interesting brands.

3.2. Attribute Scoring Functions for Numerical Attributes

Continuous values open the space for advanced math and statistics functionality. One branch of approaches is Quantile-based, using the rank (order) of attribute values for the calculation of scores. The majority of numerical ASF types however use the input values directly. We further sub-divide this group into Two-Point and Multi-Point, referring to the number of supporting points defined by the user. Two-point ASFs contain one linear or non-linear line segment. Multi-point ASFs contain multiple piece-wise defined line segments (and different functional behavior respectively), which are either continuous (connected) or discontinuous (disconnected).

With the differentiation between Linear, Non-Linear, Continuous, and Discontinuous ASFs, we take different MATH complexities and different requirements to VA interfaces into account.

**Two-Point, Linear**: A straightforward numerical mapping is using a linear transformation across the entire input value range. Examples include the unipolar min-max normalization, transforming values into the interval [0, 1], or the max-min normalization with an inverted valence. From our experiences gained from real-world cases, Two-Point, Linear is one of the most frequently applied types of functions. In the LineUp approach [GLG13] and the gradient brush tool in PAVED [CMMK20], such two-point functions are the default ASF. Real-world scenarios include a shopping behavior where a vacuum cleaner should have as much power as possible (positive linear) but be as cheap as possible (negative linear).

**Two-Point, Non-Linear**: Non-linear functions with two points have one line segment with a user-definable curvature. While MATH support for non-linear behavior exists [GLG13, BV85], this goes beyond existing VA capabilities. Support for non-linear ASF widens the design space considerably, as users now can steer the skewness of distributions from positive (e.g., logarithmic or square root norm), to negative (quadratic norm or exponential functions). Application examples include linearization attempts for exponential attribute distributions, such as the market values of soccer players.

**Multi-Point, Continuous**: Multi-point opens the space for piece-wise defined functions; continuous requires that all functions are connected. VolumePro [KG01] uses for example roof functions or trapezoid functions as transfer functions for their data. Another example is the utility function consisting of a convex and a concave part [Mar52] that is, e.g., used in utility theory. For a car dataset, an example could be a ramp function, as it typically arises when a user has a linear preference for cheaper cars and an upper limit for cars that are just too expensive.

**Multi-Point, Discontinuous**: With discontinuous ASFs, users can define discontinuous MATH behavior. This allows the implementation of step-functions such as a ceil or floor function. Considered visually, a supporting point is used as a splitting point for two line segments (piece-wise functions) that do no longer share the same score value. Using the car engine size in a tax assessment scenario, we have observed a user who assigned maximum scores to 1.993 cm$^3$ and 2.991 cm$^3$ engines but low scores to 2.000 cm$^3$ and 3.000 cm$^3$. The rationale was unexpected: cars are taxed based on engine size, which is why values slightly below thresholds are preferable.

**Quantile-based**: Quantile-based scoring functions borrow the MATH concept of quantile normalizations, which map elements of an input domain to values according to their quantile rank in the value distribution. Quantile-based scoring functions are useful to increase the resolution of output scores for particularly dense regions of the input space and are, e.g., used in bioinformatics to normalize microarray [AC01] or microchip [BI03] data. Quantile-based ASFs are also quite useful for attributes with outlier values, as outliers create no harm to output value distributions.

4. Tools for the Creation of Attribute Scoring Functions

We take advantage of the descriptive power of the proposed taxonomy and structure tools for the ASF creation by different types of ASFs. Again, we consider both the MATH and the VA perspective:

- **m**: baseline MATH approaches allowing users with a programming background to define preferences formally, leading to the creation of ASFs in a non-visual way
- **v**: inspiring VA approaches enabling large user groups to define attribute transformations interactively, but not applied in approaches for the creation of ASFs directly
Numerical flow for the creation of ASFs as presented in Figure 2. In practice, we discuss future work along five steps of a conceptual VA work-

tool either support categorical or numerical variables, but hardly attribute values. Finally, we observe the pattern that ASF creation support for MATH, but offer VA support for medical data instead of ASFs. HDR VolVis [YNCP06] and VolumePro [KG01] also have active interface as well as a scripting interface for the creation of covers the largest part of the design space of the taxonomy, as it directly create ASFs (“∗”).

4. Conclusion

We have presented a hierarchical taxonomy of attribute scoring functions and a tabular overview of tools enabling users to create attribute scoring functions, both covering the body of related work from a math and a visual analytics perspective. We believe that the taxonomy can guide designers of future visual analytics systems, e.g., towards the gaps of missing interactive solutions. Future work will shift focus on the generative power of the taxonomy and includes additional prototypes and implementations of visual interfaces for scoring functions, in combination with real-world data and collaborations with users. In addition, we plan to extend the visual analytics pipeline for downstream attribute weighting, e.g., to support multi-criteria optimization or complex item ranking.

the ASF creation often forms one step in longer cascades [Fek13] of VA processes. The first step is preprocessing to address data quality issues early in the process [RD00, KHP∗11, KPHH12], e.g., by using data wrangling support [KPHH11, KBM21] for multivariate or tabular data to transform attribute values into formats that are a) usable by downstream models and b) useful for a given analysis goal [BHR∗19]. With usable attribute values at hand, users can create ASFs interactively in a second step, e.g., by using LineUp [GLG∗13] as the most comprehensive VA solution observed in the literature. However, as indicated in Sections 3 and 4, there is still a lack of visual interfaces to support the full space of all ASF types as described in the taxonomy. Future work also includes systematic research into different (design) solutions for ASF creation tools. A third step addresses the weighting of attributes [WDC∗18, YH95], relevant for multi-attribute goals like optimization or ranking. Also, a fourth step is the combination of output scores, e.g., to arrive at a summary ranking of items [GLG∗13, BV85] or the assessment of alternatives [CMMK20] according to user preferences. One possible alternative to these steps is creating multi-attribute value functions [CL04], forming another direction for future work. Finally, the VA workflow leads to the analysis goal downstream, such as creating and analyzing rankings, multi-attribute optimizations, or similarity models.

**Item Filtering:** We consider item filtering a related, but a conceptually different concept. With filtering, items are removed from a focus set, whereas ASF may only de-emphasize the relevance of items. Filtering is typically supported with range sliders for VA tools [vv14, BHZ∗20], dynamic queries [AS94], or other types of faceted search [Hea09] interfaces. In contrast to LineUp [GLG∗13], we argue that filtering may be combined with tools for the creation of ASFs but does not need to be an intrinsic part of the interface. However, future work includes the study of an additional filtering step in applications using (sets of) ASF.

5. Discussion: Integration into Visual Analytics Workflows

We discuss future work along five steps of a conceptual VA workflow for the creation of ASFs as presented in Figure 2. In practice,

Table 1: Structured overview of tools for the creation of ASFs (lines) according to the types of ASFs of the taxonomy (columns). We use code “m” to mark approaches created with MATH support. Further, we use “v” to refer to inspiring VA approaches, which however only allow the creation of functions related to ASFs. Finally, with “x”, we mark VA approaches with a perfect fit.

<table>
<thead>
<tr>
<th>Attribute Numerification [JFJ08]</th>
<th>Categ.</th>
<th>Numerical</th>
</tr>
</thead>
<tbody>
<tr>
<td>Score Assignment</td>
<td>Explicit</td>
<td>Non-Equidistant</td>
</tr>
<tr>
<td>LineUp: Data Mapping Editor [GLG∗13]</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>PAVED [CMMK20]</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>uRank [dSSV15]</td>
<td>v</td>
<td></td>
</tr>
<tr>
<td>RanKit [KVD’18]</td>
<td>v</td>
<td>v</td>
</tr>
<tr>
<td>ValueChart [CL04]</td>
<td>v</td>
<td></td>
</tr>
<tr>
<td>Podium [WDC∗18]</td>
<td>v</td>
<td></td>
</tr>
<tr>
<td>HDR VolVis [YNCP06]</td>
<td>v</td>
<td></td>
</tr>
<tr>
<td>VolumePro [KG01]</td>
<td>v</td>
<td></td>
</tr>
<tr>
<td>TOPSIS [YH95]</td>
<td>m</td>
<td></td>
</tr>
<tr>
<td>Weighted Sum [MA10]</td>
<td>m</td>
<td></td>
</tr>
<tr>
<td>SMARTER [EB94]</td>
<td>m</td>
<td>m</td>
</tr>
<tr>
<td>SMAA [TF08]</td>
<td>m</td>
<td>m</td>
</tr>
<tr>
<td>LineUp: Scripting Interface [GLG∗13]</td>
<td>m</td>
<td>m</td>
</tr>
<tr>
<td>Promethee [BV85]</td>
<td>m</td>
<td>m</td>
</tr>
<tr>
<td>Transfer Function [CKLG98]</td>
<td>m</td>
<td>m</td>
</tr>
<tr>
<td>WWW-NIMBUS [MM00]</td>
<td>m</td>
<td>m</td>
</tr>
<tr>
<td>ValueTree [CL04]</td>
<td>m</td>
<td></td>
</tr>
</tbody>
</table>

Figure 2: Conceptual VA workflow for the creation of ASFs with five steps where interactive-visual steering support is possible.
References


© 2021 The Author(s)The Eurographics Association.