Eurographics/ IEEE-VGTC Symposium on Visualization 2008
A. Vilanova, A. Telea, G. Scheuermann, and T. Möller
(Guest Editors)

Volume 27 (*2008*), *Number 3*

Towards closing the analysis gap: Visual generation of decision supporting schemes from raw data

T. May[1] J. Kohlhammer[2]

[1]Fraunhofer Institut für Graphische Datenverarbeitung, Germany
[2]Interactive Graphics Systems Group (GRIS), TU Darmstadt, Germany

Abstract

The derivation, manipulation and verification of analytical models from raw data is a process which requires a transformation of information across different levels of abstraction. We introduce a concept for the coupling of data classification and interactive visualization in order to make this transformation visible and steerable for the human user. Data classification techniques generate mappings that formally group data items into categories. Interactive visualization includes the user into an iterative refinement process. The user identifies and selects interesting patterns to define these categories. The following step is the transformation of a visible pattern into the formal definition of a classifier. In the last step the classifier is transformed back into a pattern that is blended with the original data in the same visual display. Our approach allows in intuitive assessment of a formal classifier and its model, the detection of outliers and the handling of noisy data using visual pattern-matching. We instantiated the concept using decision trees for classification and KVMaps as the visualization technique. The generation of a classifier from visual patterns and its verification is transformed from a cognitive to a mostly pre-cognitive task.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation - Viewing Algorithms, I.3.6 [Computer Graphics]: Methology and Techniques - Interaction Techniques

1. Introduction

In this paper we introduce a concept for the combination of statistical data classification and interactive visualization. Statistical data classification techniques generate mappings that group data items into categories. These mappings are based on characteristic attributes of the items and based on a training subset of items, whose classification is initially known. The first role of visualization in our concept is the interactive definition of this training set. A user identifies interesting patterns in a visualization for the initial classification. A single pattern corresponds to a subset of data items that constitute one category. At least one other category is given by the complementing set, though the concept as such is not restricted to just two categories.
The second role of visualization is the intuitive validation of the classifier. A valid classifier must at the very least reproduce the training subset. We propose to add a visual overlay to the visualization technique, which is used to blend the original data, the selection of the training subset, and a visualization of the categories as represented in the classifier function.
By combining classification and visualization techniques, we extend the information visualization pipeline by Card [CMS99] and establish an iterative cyclic process (see Figure 1) with the goal of iteratively refining the classifier under the control of the user. A single cycle starts with the selection of data in the display, proceeds with the update of the classifier, and ends with the visual feedback of the new categories. Closing the loop, the user may proceed with modifying his selection based upon the visual feedback.
By analysis gap, we refer to the problem of the derivation, manipulation and verification of analytical models from raw data. In our case, a classifier is generated in a visual matchmaking process between the patterns in the original dataset and the patterns reproduced by the classifier. The visual matchmaking eases tasks such as:

- Validation of the appropriateness of the classifier model
- Visual separation of the pattern from the noise
- Identification of outliers or regions in the dataset, which are not covered by the classifier function

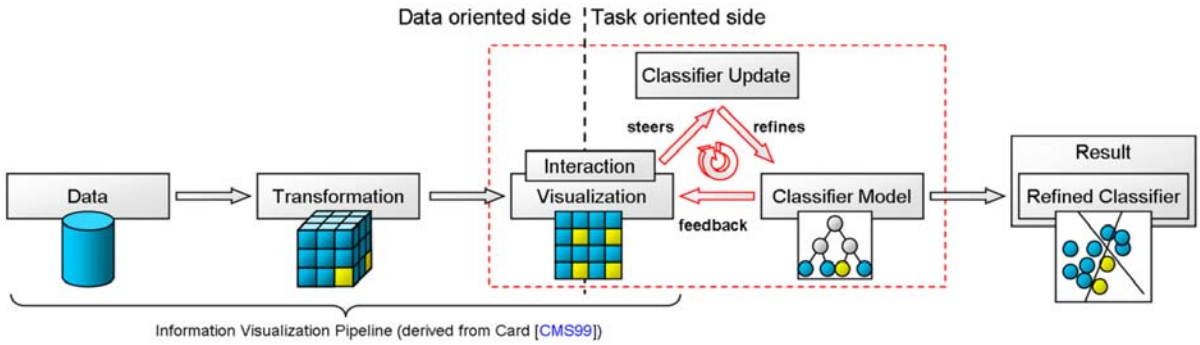

Figure 1: *We extend the information visualization pipeline by a process (red) that iteratively refines the instance of a classifier model as a reaction to user input. An iteration starts with the selection of a training classification, proceeds with the update of the classifier and ends with the combination of the aggregate data and the classifier mapped into the same visualization space. The motivation of this approach is to allow an easy match between the visual/mental model and the formal model for the classification. The classifier can be used to compare the relevance of attributes in the search for hidden dependencies in a multi-dimensional dataset.*

- Manual trade-off of precision versus recall in the classifier

The combination of techniques for the automatic generation of classifiers, and the integration of a visual interface for intuitive control and feedback gives us the opportunity to exploit the synergies between individual techniques. The specific yet distinct strengths of human and machine are connected in the iterative process.

To describe the techniques used for classification and for visualization and to verify our approach, we instantiate the concept using decision trees as a classifier model and the modified Karnaugh-Veitch-Maps (KVMaps) (see [May07]) as a visualization technique. As a specific classification scenario, we search for hidden dependencies between individual attributes in a multi-dimensional dataset. Decision trees represent a simple yet important subset of all possible dependencies.

We use the KVMaps because we assume that the complexity of the dependencies may span an arbitrarily high number of dimensions. The technique trades off two competing issues: the presentation of data from as many dimensions as possible, and the guarantee that the design of the visualization does not imply or prefer relations between data attributes simply because they are mapped to specific visual attributes.

After introducing related concepts and techniques in Section 2, we will introduce the terms and formal definitions used throughout the paper in Section 3. Section 4 constitutes the technical core of the paper, describing the process cycle of the iterative refinement of the classifier. The subsections 4.1, 4.2 and 4.3 each describe one step of this cycle: Interactive visualization, classification and visual feedback, while section 4.4 is a technical comparision of the techniques used for visual feedback. Section 5 covers a first evaluation of the concept and the implemented prototype in the context of prospective real-life scenarios. The final Section 6 wraps up the scientific contribution of the paper and gives an outlook to possible further developments.

2. Related Work

In line with the idea of Visual Analytics as proposed in the research agenda [TC05] we combine methods from statistical analysis and interactive visualization in order to support the generation of information or knowledge across different levels of abstraction. Most similar to our work is the concept recently introduced by Keim et al. [KMS*07]. They proposed the manual selection of "areas of interest" and analyzed their characteristics in order to rearrange the visual layout. The layout emphasizes the relation between the selected portions of the data space and the relevance of the corresponding values. In contrast to this approach, our primary goal is the progressive refinement of an internal classifier model. We combine the visualization of the original data and the classifier in the same visual display.

Other visual approaches to find general classifiers were proposed by Yang et al. [YWRH03] and Bendix [Ben06]. Yang et al. applied a semi-automatic clustering technique for the data dimensions. Even though different abstraction levels were provided to control the analysis, additional cognitive effort is needed for the required transformation between the abstraction levels. The *parallel sets* introduced by Bendix provide a data-abstraction technique that is based upon the well-known *parallel coordinates* approach for multi-dimensional data. Interestingly, the classification was done in the same visual space as was provided for the data, allowing for a good match-making.

Such a semi-automated analysis process is not restricted to categorical data. Heine et al. [HS07] introduced a technique for semi-automated graph clustering based on visual inter-

action. Aside from the visualization, the main difference between our technique and the ones presented there is the explicit instantiation of the classifier model for the data and its instant replication in the visualization.

Our visualization technique can be classified as a *mosaic-based* layout as presented by Hofmann et al [HSW00] [UTH06] that investigates different visual attributes for the display of the data, including different grey shades, shapes and sizes of the visual elements. The technique introduced the visualization as a mining technique for *association rules* that could be regarded as the counterpart to the classifier model, but their approach does not intuitively support the verification of these rules. Moreover, the technique can be classified as a *matrix-based* recursive layout, which were discussed in general by Keim et al [KAK95]. Langton et al. [LPWH06] presented a specific application of this in the field of neurosciences.

On a technical level, Kosara et al. [KMH*02] contributed with their rules for *blur* as a visual attribute in focus+context visualization. We suggest blur as one technique to highlight the instance of the data model in the visualization but we use a different method using the graphics hardware for rendering as proposed by Chia et al. [CCAD01].

We use convexity as the second method to highlight the classifier. It is inspired by the work of Forsell et al. [FSL05], even though we do not use spatial geometry but shading to invoke the impression of convexity.

3. Definitions

We consider a set S of data items of the same type. This set is the training set for the classifier. Every item is represented as a vector of attributes $s = (s_1, s_2, s_3, , s_n)$, with n being the number of dimensions in the dataset. In the raw data format, an attribute can be nominal, ordinal or numerical. For the KVMap, we want to treat all attributes in an identical fashion. Because of this, the KVMap discretizes the n-dimensional space along its main axes into a number of partitions. For a detailed description of how this partitioning works, we refer to [May07].

For a formal definition of the classification task, we define a *target set* T. In general, the target set could be an arbitrary subset of the complete dataset S. In the usual case, the target set is defined as the set of items where one or more attributes each lie in a specific range of values. These attributes become the dependent attributes for the classification. All other attributes become the independent attributes then the classification task can be defined as the search for a classifier function, which separates the target set from its complement T^c:

$$classifier(s) = \begin{cases} true : s \in T \\ false : s \in T^c \end{cases} \quad (1)$$

Only independent attributes are considered for the classifier function. In this paper the classifier model is a decision tree. We define the decision tree as a set of nodes V, connected to a number of child nodes. Every inner node of the tree does a classification of the data items with regards to the value of exactly one independent attribute. This classification is defined by a partitioning of the range of values of this attribute. The partitioning is a function $\Phi : S \rightarrow \wp(S)$ (the power set of S). Every partition corresponds to a child node and the result of the classification is the mapping of a data item to the child node.

The root node classifies the complete dataset. As an important consequence, all nodes $v \in V$ are recursively associated with a set $S_v \subseteq S$. S_v contains all data items, that satisfy all classifications along the path between a node v to the root node.

The leaf-nodes define the result of the classifier function for their associated subset S_v. With an optimal classifier the following equivalence applies for every leaf-node v:

$$classifier(S_v) = \begin{cases} true \Leftrightarrow : S_v \subseteq T \\ false \Leftrightarrow : S_v \subseteq T^c \end{cases} \quad (2)$$

With noisy data, it is unlikely that all leaf-nodes will allow a perfect separation of the target set and its complement. For every leaf-node the number of items in the target set and the number of items is stored, instead of the Boolean value. It is not reasonable to define a threshold value based upon the ratio $q = \frac{|T \cap S_v|}{|S_v|}$ a priori, because its semantics is closely connected to the application. The quality of a leaf-node in terms of the separation will be measured by its binary entropy measure:

$$H(T, S_v) = -q \cdot log_2(q) - (1-q) \cdot log_2(1-q) \quad (3)$$

The measure combines the distributions of T and T^c in the leaf-nodes of the decision tree. That way, we keep the information about the areas of the dataset, which are classified correctly and we keep the information about the areas, where the classifier does not separate T and T^c sufficiently well.

4. The Iterative Refinement Process

This section describes the coupling of the visualization technique with the decision tree for the data classification. The subsections follow the path of the iterative cycle. The first part of the cycle defines the input for the definition of the classifier function. The second part of the cycle describes how we create the classifier function and how we modify a method for tree-pruning to get information about the classifier, which can be remapped into the visualization space. In the last part, we discuss three methods for the blending of the original data and the classifier.

4.1. Visualization

The KVMaps are used to define the classification of the training set interactively. The visualization technique uses a multi-dimensional partitioning of the dataset, which is arranged in a two-dimensional recursive layout (see Figure

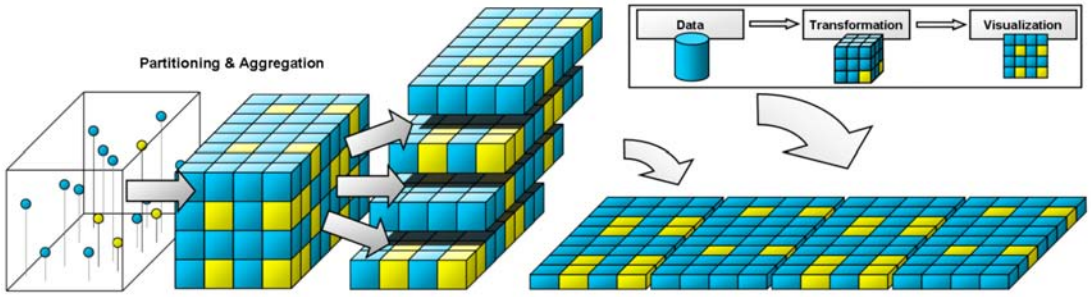

Figure 2: *This image sketches the visual mapping of the data-(hyper-)cube in the modified Karnaugh-Veitch Layout. Following the data aggregation process (left), the cube is sliced along one dimension and its partitioning. The slices are laid out along the horizontal and vertical axes (right). As the cube looses one dimension by this process, it can be repeated until all dimensions are mapped onto two-dimensional space. The information about the spatial coherency in the higher dimensions is not completely lost: The visualization exploits the ability of the human eyes to track different spatial frequencies, each of which represents one of the original dimensions. In this example, the new attribute (from third dimension) is mapped to a horizontal frequency of four, denoting that rectangles having that distance are closely related (in terms of the third attribute).*

2). Every dimension of the dataset selected for visualization is mapped onto two-dimensional space. Depending on their ordering, the dimensions are mapped on distinct spatial frequencies. In the resulting image, features of this frequency indicate the relation between the data partitions in higher dimensions. The KVMaps display aggregate information about every individual partition P using a color map with near-constant brightness. Currently we have three types of aggregates to choose from (see Figure 3):

- $|P|/|T| \,(=: E_T)$
- $|P|/|S| \,(=: E_S)$
- $\left(E_S \cdot \frac{|T|}{|S|} - \frac{|P \cap T|}{|S|}\right) / \sqrt{\frac{|T|}{|S|} \cdot (1 - \frac{|T|}{|S|}) \cdot E_S \cdot (1 - E_S)}$

The first two aggregates give information about the distribution of the data items in S and T across the partitions shown. If these distributions are independent, the third aggregate (the *empirical correlation coefficient*) will be zero in all partitions and no information can be gained for the classifier. Otherwise, specific partitions will show a positive or negative correlation for items belonging to the target set. Repetitive color patterns of partitions showing the same correlation or tendency are an indication for an underlying relation between one or more independent attributes and the target set.

The next step in the iterative process is the formalization of such a pattern. To start the iteration, the human user has to select or deselect a partition. The idea is to transfer the information about the "mental image" to the machine. All selected partitions define the initial classification of the training set and can be regarded as an approximation of T. We denote T_k as the union of all partitions selected in an iteration cycle k. Every single interaction forces an update to the classifier model, which is described in the following section.

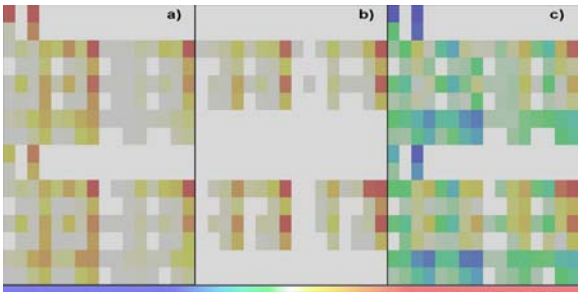

Figure 3: *This image shows the three types of aggregate data currently supported in the KVMap. Every rectangle refers to a partition: a subset of data items sharing a specific profile defined with independent attributes. Figure* (a) *shows the relative distribution of all items (E_S), red indicating common profiles, light grey indicating rare profiles and white indicating empty partitions. Figure* (b) *shows the relative distribution of items belonging to the target set defined by a dependent attribute (E_T). The statistical correlation coefficient* (c) *combines the former two values, with red and yellow indicating positive and blue and green negative correlations.*

4.2. Classification

The strategy for the creation of a decision tree is a representation of the set T_k with as little resources as possible. "Resources" are the average number of decisions to be made in the application of the classifier. The divide-and-conquer technique applied here is known from the field of data-mining (see, for example [HK06]). The root node represents the complete dataset. For every node v, we identify the attribute that provides the largest increase of information when used for the separation of T_k and its complement. For a quality measure we define the information necessary to clas-

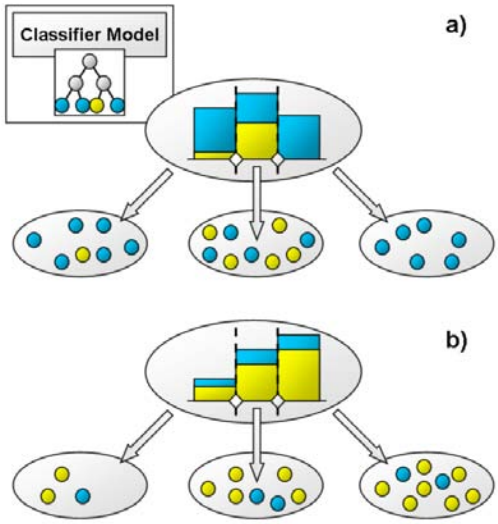

Figure 4: *The entropy measures the discriminating value of an attribute partitioning. The histograms denote the distribution of S (blue) and its subset T_k (yellow) along the value-range of an attribute. Individual partitions are shown in the child nodes. Figure* a) *shows a good discrimination (low entropy) of the subset T_k in the individual partitions, while* b) *shows a bad discrimination. An optimal decision tree minimizes the entropy in its leaf nodes with as little inner nodes (decisions) as possible.*

sify this set using the binary entropy measure $H(T_k, S_v)$ for the associated subset S_v.

Usually, the partitioning of the attribute for the KVMap is also used for the decision tree. However, if a single partition is empty, it will be removed from the tree. For the partitioning Φ of an attribute and the set S_v associated to a node, the entropy can be defined as follows:

$$H_\Phi(T_k, S_v) = \sum_{P \in \Phi(S_v)} \frac{|P|}{|S_v|} \cdot H(T_k, P) \tag{4}$$

Figure 4 shows two nodes with different entropies in the decision tree. The formula implies that partitions with identical binary entropies can be merged without loss of information. We are now able to compare the different attributes and their partitions for a given node v. The partitioning of the selected attribute recursively defines the number of child nodes and their associated subsets. The process stops when either the entropy reaches zero or no attributes are left.

To avoid over fitting the classifier, *pruning* techniques are applied in data mining. They cut the size of the tree based upon an estimated trade-off between its depth and the added information. To qualify this trade-off, an *information gain* is defined for every node in the tree. Instead of pruning the tree we use the information gain of a node v to indicate the relatedness of its associated subset S_v to the set T_k specified

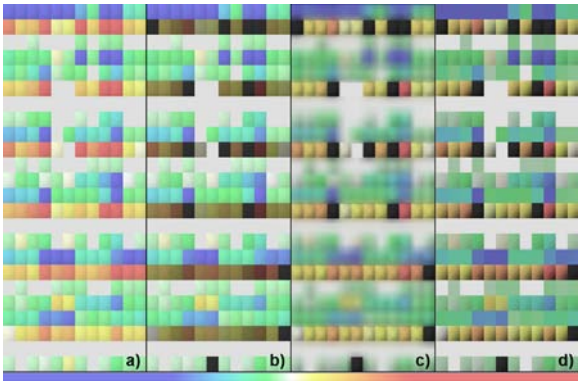

Figure 5: *This image shows a sample using the three visual attributes brightness* (b), *blur* (c) *and convexity* (d). *To compare the three visualizations the same pattern* (a) *is used with the same partitions selected. The attributes have different advantages and disadvantages: Brightness has its strengths when many dimensions and partitions are shown and the rectangles are small, but it obscures the selected pattern more than the other attributes. Blur has a small visual range; only two or three different degrees of blur can be distinguished at a glance. The three attributes can easily be switched, but convexity appears to be the attribute of choice.*

by the user. Formally, the information gain of a node v for an attribute $attr$ and its partitioning Φ_{attr} defined as follows:

$$gain_{attr}(T_k, S_v) = H(T_k, S_v) - H_{\Phi_{attr}}(T_k, S_v) \tag{5}$$

Let $v_0, v_1, v_2, \ldots v_m, v_{leaf}$ be the path of nodes from the root-node to a leaf node and let $attr_0, attr_1, attr_2, \ldots attr_m$ be the attributes defined for every inner node in the path. We compute the "measure of relatedness" σ as the weighted sum of the information gain along this path:

$$\sigma(v_{leaf}) = \frac{1}{H(T_k, S)} \sum_{i=0}^{m} gain_{attr_i}(T_k, S_{v_i}) \cdot \left(\frac{|S_{v_i} \cap T_k|}{|S_{v_i}|} \right) \tag{6}$$

The function merges the *significance* of the information unit and the actual *value* of that information. Every partition of the original data in the KVMap corresponds to exactly one leaf node. This allows us give feedback about the representation of the set T_k in the decision tree using the same visual space.

4.3. Visual Feedback

The feedback is the projection of the resulting classifier function into the visualization used for aggregate data. At this point, we have three types of information for each partition shown in the KVMap:

- The aggregate data (see Section 4.1)
- The selection status, denoting when a partition belongs to the subset T_k

- The actual feedback, denoting the likelihood of a partition to complement the subset T_k

Selected partitions must be easy to distinguish for deselection and are colored black independently of the other values. Note that the user is not required to select all partitions of a pattern. The first and the third type of information are scalar values with a predefined maximum and minimum. We blend these two types in the same visual space. Usually it is not a problem to map more than one visual attribute to visual structures (e.g. glyphs). In this case, we had to choose an attribute that does not interfere with the identification of patterns. Since the patterns manifest as horizontal and vertical color repetitions of varying frequencies, changing position, size or even only the shape of the visual structures causes interference and might be perceived as artefacts. In addition to hue used for the aggregate data, we consider three visual attributes for our prototype (see Figure 5):

- Brightness
- Focal Blur
- Convexity

Our choice is based upon three requirements: All attributes can be pre-attentively processed according to [War04], they do not need additional visual space, and they may not interfere much with hue, that is used for the aggregate data.
The adjustment of the brightness value for every rectangle is straightforward, because technically brightness can be defined along with hue. By *convexity* we mean the perceived three-dimensional shape of the rectangle, which is generated from (two-dimensional) shading. It is implemented by using four triangles to render to a rectangle for a partition. The brightness is set for every individual vertex. Shading is used in any case to visually separate neighboring rectangles of the same color. By controlling the degree of convexity, the new attribute can be introduced without large modifications.
The emulation of focal blur alone requires a little more effort. The challenge is to blur different part of the image with defined degree of blur and to do this fast enough to support real-time feedback on user interaction. We used a method presented by Chia et. al. [CCAD01], which exploits that texture mipmaps can be computed in graphics hardware allowing for fast generation of different degrees of blur. We do not rely on photorealistic blur so we expected that the box-filter will work. The rectangles are rendered in varying distances depending on the value σ. The information in the z-Buffer is used to mask the actual degree of blur when the texture mipmaps derived from the KVMap image are drawn in their corresponding distances.
The feedback closes the iteration cycle. Every selection or deselection of a partition in the KVMap changes the classifier and the feedback (see Figure 6). The coherency of the two patterns (original data and classifier) displayed in two independent visual attributes is used to estimate the quality of the approximation of the pattern. The user terminates the iterative process and may use the formal representation of the classifier to visualize the decision tree (see Figure 7) or start a detailed analysis on the basis of the relevant attributes in the dataset.

4.4. Discussion

We compared the three visual attributes brightness, blur and convexity based on the following criteria:

1. Non-interference with pre-attentive identification of the hue pattern
2. Quality in relation to the resolution of the KVMap
3. Expressiveness of gradual differences

The visual attribute that interferes most with the identification of the hue pattern is the brightness, mainly because it limits the contrast of the color map used for the aggregate data. Convexity and blur do not seem to obscure the spotting of patterns of the aggregate data.
The size of the rectangles in the display depending on the number of partitions shown influences the usability of all three attributes. Both blur and convexity are visual impressions generated in a screen area instead of a single pixel. Because of that, very small rectangles (<20 pixels) mostly inhibit the use of blur and still pose a limitation to convexity, while the perception of brightness is barely affected.
Gradual differences are necessary to convey small changes of σ, whenever the selection of partitions changes. Brightness and convexity can be visually separated into a relatively high range of values. Blur performs exceptionally weak on this property.
The number of partitions is connected to the size of the display: A typical desktop screen is able to display 10-16 dimensions. The maximum possible resolution (one pixel per partition) is almost never used, because the partitions can not be selected when the rectangles are extremely small. Focus+context techniques and the ability to the use the selected pattern to define a zoom into the dataset circumvent this problem. The attributes can be switched easily based on user preference, but convexity seems to be the method of choice for most application areas.

5. Evaluation

From the conceptual point of view, the evaluation rates the appropriateness of the methods presented here for the visual matchmaking process presented in the introduction. In public demonstrations with prospective users from fields as different as security analysis, insurances, demographics, pharmaceutical industry and medicine as well as members of the scientific community, the generic task and the prototype were discussed. In the context of multi-dimensional data analysis, the defensible identification of relevant attributes is regarded as an important prerequisite for a detailed analysis. The users understood that the results of this approach qualitatively indicate the direction of further detailed analysis and that they complement existing quantitative techniques.

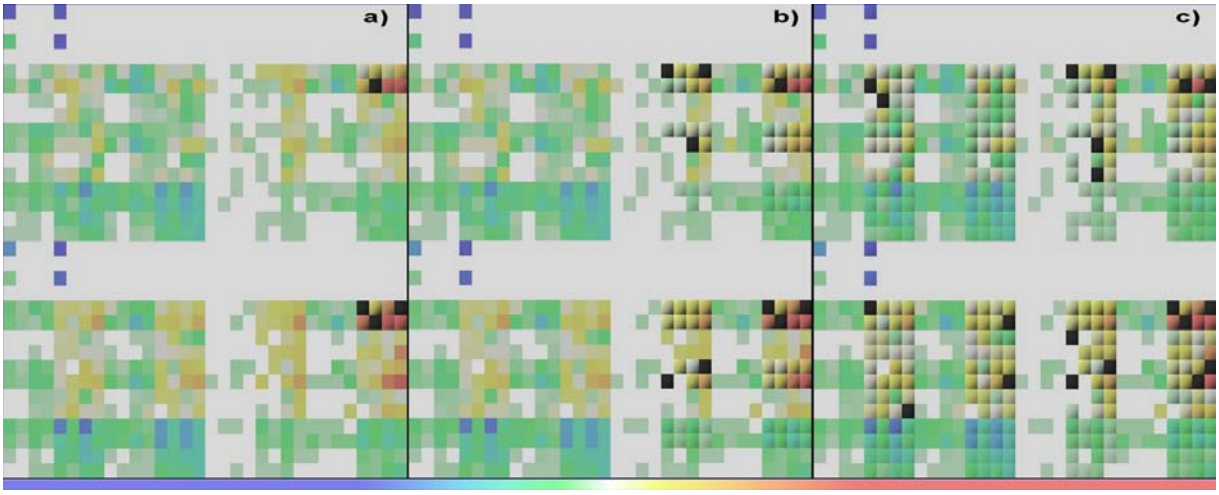

Figure 6: *This image shows the results after 4* (a), *9* (b) *and 20* (c) *refinement cycles. Each cycle corresponds to one partition selected by the user, which is colored black. The information about the partitions that are (more or less) likely to complement the visual pattern is blended with the original color using the convexity attribute.*

We tested the prototype with demographic data (public use micro census) and an online-shopping database containing additional information about hoax orderings. Even though the capabilities of people in spotting and matching patterns in the display are different, most users (>90%) were able to see patterns and to separate patterns from the noise (with a noise-to-data ratio of 2:1 and more). The majority of users were able to distinguish between different patterns and to understand the behavior of the feedback loop without intensive guidance.

One result of the discussion is the fact that *pattern identification* is a subtask different from *pattern understanding*. The KVMap is an extreme case for the focus on pattern identification, because we wanted to avoid interference between pre-cognitive and cognitive subtasks. Actually, these tasks can be performed separately, but even when people see a pattern, they have difficulties with clicking on the map without contextual information about the semantics of the pattern. The challenge of providing this context (especially for the inexperienced user) without causing interference is a topic beyond the scope of this paper.

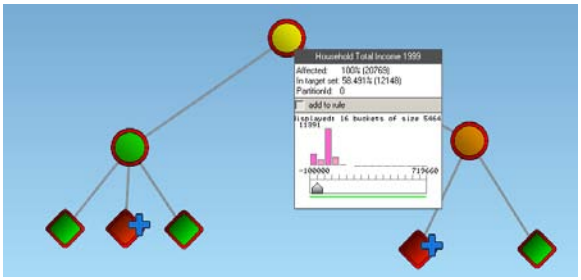

Figure 7: *After the refinement and definition of the formal model, it can easily be exposed for further processing or communication. We use this different perspective for* pattern understanding. *Colours in the tree indicate the fraction of elements in that set belonging to T_k (from green, via yellow to red). A user can manually modify and prune the tree and indicate his finalized decision scheme for T by selecting the corresponding leafnodes (blue crosses).*

6. Conclusion

We presented a concept for the combination of data classification with a visualization of multi-dimensional data for the interactive iterative refinement of a classifier. The concept is instantiated with the motivation to identify relevant attributes for the classification task. From the outset, the KVMap was a technique solely dedicated to pattern identification as opposed to pattern understanding. By the interactive refinement of a formal classifier we gained the opportunity to do both, but on terms of the pre-cognitive abilities of humans: The refined formal model can easily be transformed into a form, usable for further processing and communication, which is further step a along the analytical pipeline from data to decision.

As a concept, it formulates a new approach to combine data analysis techniques with visualization and it can be explored along at least three dimensions: We expect that the concept can be instantiated using other visualization techniques, other analytical models or different tasks (and combinations thereof). The decision trees, for example, are selected for their simplicity, but they are not expected to be the best analytical model for all types of tasks. Future work includes

the integration of other analytical models and the choice between more than one analytical model in the same prototype based on the visual matchmaking process.
With this concept we merged the visualization based upon a *data model* with a visualization based upon an *analytical model*. Visualization techniques are used to support a coherent image of the information across different levels of abstraction. This support for the transformation of data to information can be a valuable addition to the field of Visual Analytics.

7. Acknowledgements

We thank Peter Stephenson for his valuable help in the improvement of this paper.

References

[Ben06] BENDIX F.: Parallel sets: Interactive exploration and visual analysis of categorical data. *IEEE Transactions on Visualization and Computer Graphics 12*, 4 (2006), 558–568. Member-Robert Kosara and Member-Helwig Hauser. 2

[CCAD01] CHIA N., CANT R., AL-DABASS D.: New anti-aliasing and depth of field techniques for games graphics. In *2nd International Conference on Intelligent Games and Simulation (GAME-ON 2001)* (London, UK, 2001), pp. 115–. 3, 6

[CMS99] CARD S. K., MACKINLAY J. D., SHNEIDERMAN B.: *Readings in information visualization: using vision to think*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999. 1

[FSL05] FORSELL C., SEIPEL S., LIND M.: Simple 3d glyphs for spatial multivariate data. In *INFOVIS '05: Proceedings of the Proceedings of the 2005 IEEE Symposium on Information Visualization* (Washington, DC, USA, 2005), IEEE Computer Society, p. 16. 3

[HK06] HAN J., KAMBER M.: *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, March 2006. 4

[HS07] HEINE C., SCHEUERMANN G.: Manual clustering refinement using interaction with blobs. In *Proceedings of the EuroVis Conference* (2007), Eurographics Association, pp. 59–66. 2

[HSW00] HOFMANN H., SIEBES A. P. J. M., WILHELM A. F. X.: Visualizing association rules with interactive mosaic plots. In *KDD '00: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining* (New York, NY, USA, 2000), ACM, pp. 227–235. 3

[KAK95] KEIM D. A., ANKERST M., KRIEGEL H.-P.: Recursive pattern: A technique for visualizing very large amounts of data. In *VIS '95: Proceedings of the 6th conference on Visualization '95* (Washington, DC, USA, 1995), IEEE Computer Society, p. 279. 3

[KMH*02] KOSARA R., MIKSCH S., HAUSER H., SCHRAMMEL J., GILLER V., TSCHELIGI M.: Useful properties of semantic depth of field for better f+c visualization. In *Proceedings of the Joint Eurographics–IEEE TCVG Symposium on Visualization (VisSym 2002)* (2002), pp. 205–210. 3

[KMS*07] KEIM D., MORENT D., SCHNEIDEWIND J., HAO M., DAYAL U.: Intelligent visual analytics queries. In *VAST 2007: Proceedings of the IEEE Symposium on Visual Analytics Science and Technology* (Sacramento, CA, USA, 2007), IEEE Computer Society. 2

[LPWH06] LANGTON J. T., PRINZ A. A., WITTENBERG D. K., HICKEY T. J.: Leveraging layout with dimensional stacking and pixelization to facilitate feature discovery and directed queries. In *VIEW* (2006), pp. 77–91. 3

[May07] MAY T.: Working with patterns in large multivariate datasets - karnaugh-veitch-maps revisited. In *IV '07: Proceedings of the 11th International Conference Information Visualization* (Washington, DC, USA, 2007), IEEE Computer Society, pp. 277–285. 2, 3

[TC05] THOMAS J. J., COOK K. A.: *Illuminating the Path*. IEEE CS Press, 2005. 2

[UTH06] UNWIN A., THEUS M., HOFMANN H.: *Graphics of Large Datasets: Visualizing a Million (Statistics and Computing)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006. 3

[War04] WARE C.: *Information Visualization: Perception for Design*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004. 6

[YWRH03] YANG J., WARD M., RUNDENSTEINER E., HUANG S.: Visual hierarchical dimension reduction for exploration of high dimensional datasets. In *VISSYM '03: Proceedings of the symposium on Data visualisation 2003* (Aire-la-Ville, Switzerland, Switzerland, 2003), Eurographics Association, pp. 19–28. 2