Multi-Dimensional Scaling applied to Hierarchical Rule Systems

Thomas R. Gabriel Kilian Thiel Michael R. Berthold

Nycomed Chair for Bioinformatics and Information Mining
Department of Computer and Information Science
University of Konstanz, Box 712, 78457 Konstanz, Germany
Email: {gabriel,thiel,berthold}@inf.uni-konstanz.de

Abstract— *This paper presents an approach for visualizing high-dimensional fuzzy rules arranged in a hierarchy together with the training patterns they cover. A standard multi-dimensional scaling method is used to map the rule centers of the top hierarchy level to one coherent picture. Rules of the underlying levels are projected relatively to their parent level(s). In addition to the rules, all patterns are mapped onto the two-dimensional projection in relation to the positions of the corresponding rule centers. Visualization is further extended by showing hierarchical relationships between overlapping rules of different levels, which are generated by a hierarchical rule learner. This delivers interesting insights into the rule hierarchy and offers better explorative properties. Additionally, rules can be highlighted interactively emphasizing the subsequent rules at all underlying levels together with the patterns they cover. We demonstrate that this technique allows investigation of interesting rules at different levels of granularity, which makes this approach applicable even for a large number of rules. The proposed technique is illustrated and discussed based on a number of hierarchical rule model visualizations generated from well-known benchmark data sets.*

Keywords— Multi-Dimensional Scaling, Fuzzy Rule Induction, Rule Hierarchy, Rule Visualization.

1 Introduction

Rule learning algorithms are widely used in data mining to automatically extract rules from data. In [7, 15] and [20], algorithms are described that construct hyperrectangles in feature space. The resulting set of rules encapsulates regions in feature space that contain patterns of the same class. Other approaches, which construct fuzzy rules instead of crisp rules, are presented, for example, in [1, 12, 18] and [19]. All of these approaches have in common that they tend to build very complex rule systems for large data sets originating from a complicated underlying system. In addition, high-dimensional feature spaces result in complex rules relying on many attributes further increasing the number of required rules to cover the solution space. An approach that aims to reduce the number of constraints on each rule individually is presented in [3]. The generated fuzzy rules only constrain a few of the available attributes and hence remain readable even in the case of high-dimensional spaces. The fuzzy rules generated by this method have been visualized by parallel coordinates [4, 10].

In [8], we described a method that attempts to tackle this inherent problem of interpretability of large rule models. We achieve this by constructing rule models with varying degrees of complexity. The method builds a rule hierarchy for a given data set. The rules are arranged in a hierarchy of different levels of precision. Lower levels of the model describe regions in input space with low evidence in the given data, whereas rules at higher levels describe more strongly supported concepts of the underlying data. The method is based on the fuzzy rule learning algorithm described in [3, 9], which builds a single layer of rules autonomously. We use the resulting rule system recursively to determine rules of low relevance, which are then used as a filter for the next training phase. The result is a hierarchy of rule systems with the desired properties of simplicity and interpretability on each level. Experimental results demonstrated that fuzzy models at higher hierarchical levels show a dramatic decrease in number of rules while still achieving a better or similar generalization performance than the fuzzy rule system generated by the original, non-hierarchical algorithm.

In this paper we present an approach that enables the visualization of hierarchically structured rules and data in one coherent plot. A standard multi-dimensional scaling (MDS) method is applied to project rules of a certain level from the original, high-dimensional space, into a lower, usually two-dimensional space. Rules of the underlying levels are projected relatively to their parent level(s), likewise all data points are projected in relation to the positions of the corresponding rule centers. Furthermore, the visualization shows hierarchical relationships between overlapping rules of different levels as generated by a hierarchical rule learner. Due to the hierarchical nature of the induced rule system, interactive exploration becomes possible across the entire rule model and provides interesting insights into the underlying concept.

The paper is organized as follows: In the next section we briefly describe the used hierarchical rule learning method, followed by a short introduction to multi-dimensional scaling methods, and how data and hierarchical rules can be visualized by applying an extended multi-dimensional scaling method. In the following section, we describe how rule hierarchies, together with their original data, can be explored within this visualization. To illustrate the proposed method, hierarchical rule systems are generated based on the well-known iris data and on the vehicle silhouettes data.

2 Learning Hierarchical Rule Systems

The rule induction algorithm used here is based on a method described in [3], which is based on an iterative algorithm. During each learning epoch, i. e. presentation of all training patterns, new fuzzy rules are induced when necessary and existing ones are adjusted whenever a conflict occurs. For each pattern three main steps are executed. First, if a new training pattern lies inside the support-region of an existing fuzzy rule of the correct class, its core-region is extended in order to cover the new pattern. Second, if the new pattern is not yet

ISBN: 978-989-95079-6-8

covered, a new fuzzy rule of the correct class is introduced. The new example is assigned to its core, whereas the support-region is initialized by infinity, that is, the new fuzzy rule covers the entire domain. Finally, if a new pattern is incorrectly covered by an existing fuzzy rule, the fuzzy rules' support-region is reduced to avoid the conflict. This heuristic for conflict avoidance aims to minimize the loss in volume. In [9], three different heuristics are compared to determine the loss in volume. As discussed in [3], the algorithm terminates after only a few iterations over the set of example patterns.

The final set of fuzzy rules can be described as an n-dimensional IF clause as antecedence and one assigned *class* in the rule's conclusion:

$$\begin{array}{llllll}
\mathcal{R}_1^1 & : \texttt{IF} & x_1 \,\texttt{IS}\, \mu_{1,1}^1 & \wedge \cdots \wedge & x_n \,\texttt{IS}\, \mu_{n,1}^1 & \texttt{THEN class } 1 \\
 & & \vdots & \vdots & \vdots & \\
\mathcal{R}_{r_1}^1 & : \texttt{IF} & x_1 \,\texttt{IS}\, \mu_{1,r_1}^1 & \wedge \cdots \wedge & x_n \,\texttt{IS}\, \mu_{n,r_1}^1 & \texttt{THEN class } 1 \\
 & & \vdots & \vdots & \vdots & \\
\mathcal{R}_j^k & : \texttt{IF} & x_1 \,\texttt{IS}\, \mu_{1,j}^k & \wedge \cdots \wedge & x_n \,\texttt{IS}\, \mu_{n,j}^k & \texttt{THEN class } k \\
 & & \vdots & \vdots & \vdots & \\
\mathcal{R}_{r_c}^c & : \texttt{IF} & x_1 \,\texttt{IS}\, \mu_{1,r_c}^c & \wedge \cdots \wedge & x_n \,\texttt{IS}\, \mu_{n,r_c}^c & \texttt{THEN class } c
\end{array}$$

where $\mathcal{R}_j^k$ represents rule j for class k. The rule base contains rules for c classes and r_k indicates the number of rules for class k ($1 \leq j \leq r_k$ and $1 \leq k \leq c$). The fuzzy sets $\mu_{i,j}^k : \mathbb{R} \mapsto [0,1]$ are defined for every feature i ($1 \leq i \leq n$), but in cases of unconstrained features the membership degree constantly remains 1. The overall degree of fulfillment of a specific rule for an input pattern $\vec{x} = (x_1, \ldots, x_n)$ can be computed using the fuzzy set operator for conjunction, called T-norm ($\top$):

$$\mu_j^k(\vec{x}) = \top_{i=1,\cdots,n} \left\{ \mu_{i,j}^k(x_i) \right\}.$$

The combined degree of membership for all rules of class k can be calculated using the fuzzy set operator for disjunction, called S-norm ($\perp$):

$$\mu^k(\vec{x}) = \perp_{j=1,\cdots,r_k} \left\{ \mu_j^k(\vec{x}) \right\}.$$

From these membership values the predicted class k_{best} for an input pattern $\vec{x}$ is derived as:

$$k_{\text{best}}(\vec{x}) = \arg\max_{k=1,\ldots,c} \left\{ \mu^k(\vec{x}) \right\}.$$

The algorithm uses trapezoidal membership functions, which can be described by four parameters $\langle a_i, b_i, c_i, d_i \rangle$, where a_i and d_i define the fuzzy rule's support-, and b_i and c_i its core-region for each attribute i of the input dimension. The core-region is defined as a rectangular area with an activation of 1, whereby the support-region decreases linearly to its boundaries with a degree of fulfillment of 0.

The resulting set of fuzzy rules can then be used to classify new patterns by computing the overall degree of membership for each class. The accumulated membership degrees over all input dimensions and across multiple rules are calculated by a fuzzy norm. For the purpose of this paper, we concentrate on the rules' core-regions only, that is, the part of each rule where the degree of membership is equal to 1 – resulting in crisp rules – is considered.

An extension of this rule learning algorithm is proposed in [8], which allows the generation of a hierarchy of rules. Therefore, the classical fuzzy rule algorithm is used to determine rules of low relevance recursively, which are then used as a filter for the next training phase. Training examples that cause the creation of small, less important rules are therefore excluded from the training phase of the next layer, resulting in a more general rule system, ignoring the withheld, small details in the training data. An example of a two-stage rule induction process is illustrated in Fig 1.

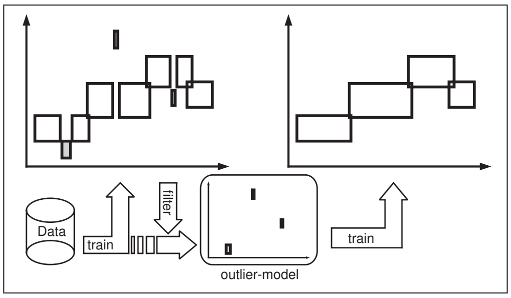

Figure 1: Two-stage outlier filtering as described in [2].

This process can be repeated to generate the desired hierarchy of rule systems with an increasing generality towards higher levels. The rule layers are arranged in a hierarchy of different levels of precision. Lower levels of the model describe regions in input space with low evidence in the training data, whereas rules at higher levels describe more strongly supported aspects of the underlying data.

3 Multi-Dimensional Scaling

To visualize objects of high-dimensional feature spaces, multi-dimensional scaling methods [6] can be applied to map them onto lower-dimensional spaces with usually two or three dimensions. In order to avoid a loss of proximity information, MDS methods try to preserve the pairwise distances between objects during the mapping process by minimizing an appropriate error function. The reduction of dimensions allows the visualization of high-dimensional objects in a lower-dimensional space, using traditional methods such as scatter plots or scatter matrices.

One technique to apply this kind of mapping by minimizing a particular error function is the well known Sammon algorithm [17]. For each object $\vec{X}_i$ in high-dimensional space $\mathbb{R}^H$, a spatial representation $\vec{x_i} \in \mathbb{R}^L$ in low-dimensional space (usually $L = 2$) has to be computed ($1 \leq i \leq N$ with N as the number of objects).

To approximate the distance information, the position of each object in the low-dimensional space has to be adjusted such that the pairwise distances between each two objects $(\vec{x_i}, \vec{x_j})$ in the low-dimensional space, $d_{ij} = d(\vec{x_i}, \vec{x_j})$, approximate the distances of the two corresponding objects $(\vec{X}_i, \vec{X}_j)$ in the high-dimensional space, $D_{ij} = D\left(\vec{X}_i, \vec{X}_j\right)$:

$$\forall_{i \neq j} : D_{ij} \approx d_{ij}, 1 \leq i, j \leq N.$$

Usually the Euclidean metric is used to measure these distances in the low-dimensional space, and often in the high-

ISBN: 978-989-95079-6-8

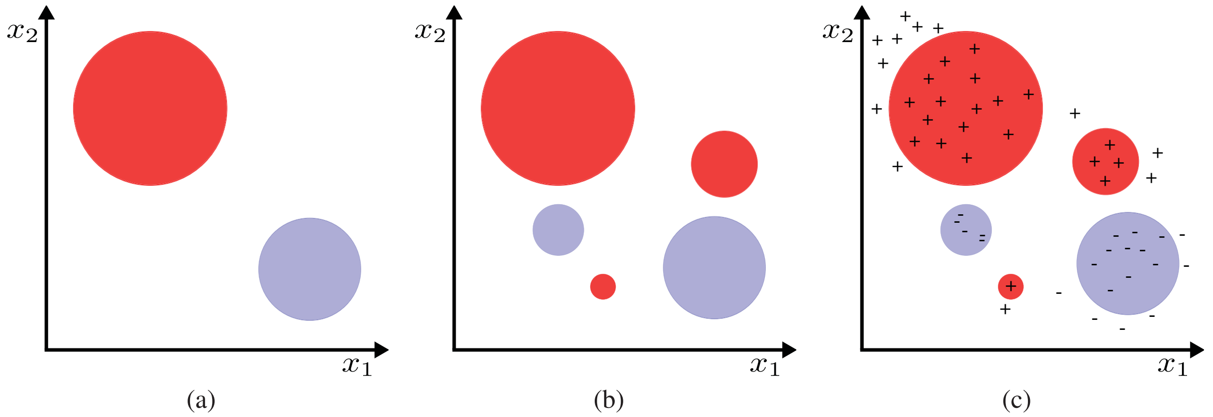

Figure 2: Shows a two-level rule hierarchy. (a) Two top-level rules assigned to two different classes, $+$ (red) and $-$ (blue), (b) three additional rules on the next level, (c) all rules together with their underlying data points.

dimensional space as well:

$$d_{ij}^2 = \sum_{l=1}^{L} (x_{i,l} - x_{j,l})^2.$$

Approximation of the pairwise distances in high- and low-dimensional space is formulated by Sammon as a minimization problem of a cost function, which aggregates the weighted squared differences of the distances in high- and low-dimensional space:

$$E = \sum_{i=1}^{N} \sum_{j>i}^{N} \omega_{ij} (d_{ij} - D_{ij})^2.$$

where the factors ω_{ij} are introduced to weight the distances individually and normalize the stress function E in order to be independent from the absolute values D_{ij}. The steepest gradient method is applied to minimize iteratively a cost function E for each object $\vec{x_i}$ at each step. Usually several iterations are needed by the algorithm to converge to a local cost minimum.

3.1 Projective Multi-Dimensional Scaling

As mentioned above, the original Sammon algorithm tries to preserve the pairwise distances of all objects. This means that the position of each object in the low-dimensional space is adjusted iteratively according to the position of the other objects in the high- and low-dimensional space. Subsequently this leads to a change of the positions of all objects in the low-dimensional space. With our projective multi-dimensional scaling approach, the positions of the objects in the low-dimensional space are adjusted according to a set of fixed projection objects.

First a set of objects in the high-dimensional space is selected. For each of these objects $\vec{X_p} \in \mathbb{R}^H$ a spatial representation $\vec{x_p} \in \mathbb{R}^L$ $(1 \leq p \leq M$ with $M \ll N$ as the number of fixed projection objects $\vec{x_p}$) has to be found by means of the standard MDS algorithm described above. Once these objects have been mapped onto the low-dimensional space, they are used as fixed objects and are not changed anymore. Furthermore, each of the other objects $\vec{X_i} \in \mathbb{R}^H$ are mapped onto the low-dimensional space according to the fixed projection objects. Therefore, the distance between a regular object $\vec{x_i} \in \mathbb{R}^L$ and a fixed projection object $\vec{x_p}$ in the low-dimensional space $d_{ip} = d\left(\vec{x_i}, \vec{x_p}\right)$ has to be approximated to the distance between the two corresponding objects $(\vec{X_i}, \vec{X_p})$ in the high-dimensional space $D_{ip} = D\left(\vec{X_i}, \vec{X_p}\right)$, which translates to:

$$\forall_{i \neq p} : D_{ip} \approx d_{ip}, 1 \leq i \leq N, 1 \leq p \leq M.$$

Again, a cost function E_P is defined that aggregates the weighted squared differences of the distances in the high- and low-dimensional space:

$$E_P = \sum_{i=1}^{N} \sum_{p=1}^{M} \omega_{ip} (d_{ip} - D_{ip})^2.$$

The projective MDS method is useful if a set of objects has to be mapped according to an already existing set of mapped objects without modifying the mapping of the latter.

In the case of high-dimensional hierarchical data, projective MDS can be used to determine a spatial representation in several iterations. First the standard MDS method is applied to the top-level data. Once this data is mapped, projective MDS can be applied stepwise to the data of the lower levels using the data of higher levels as fixed projection objects. The low-dimensional representation of the already mapped data does not change anymore when applying projective MDS to the remaining levels.

3.2 Hierarchical Rules in Multi-Dimensional Scaling

It is more complex to visualize rule models on lower dimensions than simple data points. The main challenge is the mapping of the rules' antecedences, which are usually hyperrectangles in the original high-dimensional feature space. In order to visualize high-dimensional fuzzy rules by means of MDS, the center points of a rules' core-regions are mapped onto a low-dimensional space, as described in [11, 16]. For hierarchical fuzzy rule systems, the center points of the top-level rules are mapped onto the low-dimensional space using standard MDS. Subsequently, the rules of the underlying levels are mapped level by level via the projective MDS with the parent rule centers as fixed projection points. Finally, the data points are mapped according to all projected rules, whereas the neighborhood of rules and data points is approximated by the proposed MDS. The overlapping relation between rules of different levels may get lost in the visualization, but is again taken into account when rule systems are visually explored by interactive highlighting, see Sec. 4.2 for an example.

In addition to the rules' center points, the spread of a fuzzy rule has to be mapped onto the low-dimensional space in order to visualize the rules' sizes, possible overlaps, and the coverage according to the data points the rules are based on. In [11], we visualize the overlap of rules of flat rule systems. Dealing with hierarchical fuzzy rules, we only focus on the visualization of the spread and the number of covered data points of a rule as well as the rules' level and not on overlap with other rules.

The spread is visualized by a sphere around the mapped rule center with a radius $r_i = \rho(\lambda\omega_i + (1-\lambda)\sigma_i^2)$ according to the variance σ_i^2 and the number of the points ω_i covered by a rule $\mathcal{R}_i$, with λ as weighting coefficient $\lambda \in [0,1]$ and $\rho(\cdot)$ as scaling function. The number of vertices $v = 2^H$ of a hyperplane grows exponentially with the number of dimensions H whereas all vertices have the same distance to the center point. All vertices of a high-dimensional hyperplane are mapped by MDS, representing vertices of a low-dimensional polygon; all with the same distance to the polygons center point. With increasing dimensionality, the polygon spanned by the projected vertices converges to a sphere.

In Fig. 2 (a), two top-level rules are shown, covering data points of two different classes. The red rule covers data points of class $+$, while the blue covers data points of class $-$. Fig. 2 (b) illustrates rules of the primary and the secondary level, which consists of three smaller rules, two rules covering data points of class $+$ and two of class $-$. Fig. 2 (c) shows rules of all levels as well as the data points. It can be seen that the spheres representing the rules do not necessarily cover all the data points in the low-dimensional space as they do in the high-dimensional space. This is because the radius is based on the number of covered data instances of a rule as well as the variance of the covered data points in the high-dimensional space, but is not based on the position of the data points in the low-dimensional space.

4 Visualization Hierarchical Rules and Data

The following section illustrates the proposed approach on the well-known iris dataset before looking at a larger dataset, the vehicle silhouettes data. We compare the classical, non-hierarchical rule model to the hierarchical structured rules by visualizing all the rules together with their originating data instances in one coherent picture.

4.1 Iris Dataset

The first example shows a small two-level hierarchy trained on the iris data, which consists of 150 four-dimensional patterns assigned to three classes. Fig. 3 (a) shows the top-rule level containing three rules – one for each class, (b) the top and bottom-rule levels with additional 11 rules (some cover only a single or a few instances and therefore appear as small points), and (c) all data points and all rules together. The top-level rules are projected into the low-dimensional space based on each center point. The second level is then projected based on the first rule level. In the last step, data points are projected according to both hierarchical levels as shown in Fig. 3. The bottom picture shows rules and data instances of all three classes, which are almost perfectly separated from each others by three large rules. Smaller rules in-between are needed to cover artifacts and details in the data.

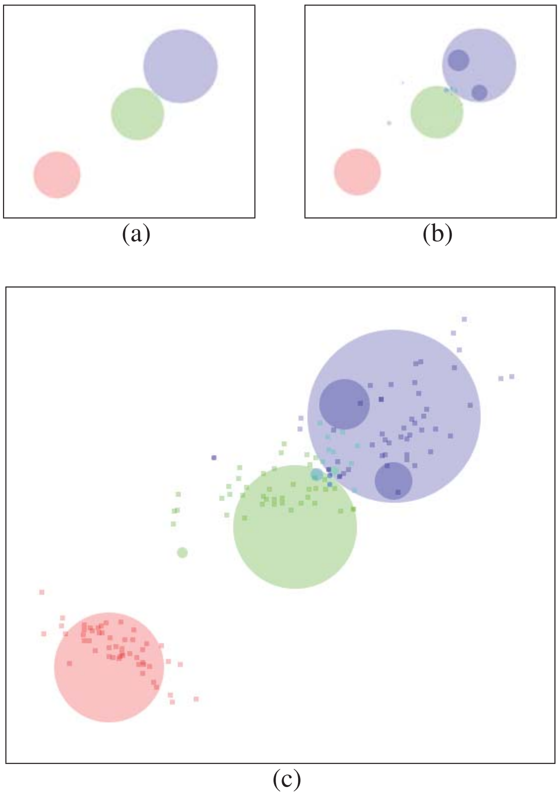
(a)
(b)
(c)

Figure 3: Two-level rule hierarchy trained on iris data: (a) three top-most rules, (b) additional 11 rules from the bottom-level, and all data points in (c).

4.2 Vehicle Silhouettes Dataset

The vehicle silhouettes dataset consists of 846 samples belonging to four car classes – Opel, Saab, bus, and van – represented in an 18-dimensional feature space. To demonstrate the usefulness of our proposed method, we trained a three-level fuzzy rule hierarchy on this data from the European StatLog–Project [14]. In a first experiment, a classical rule model with 222 rules without hierarchy information is trained on the vehicle data. As can be seen in Fig. 4, exploration is hardly pos-

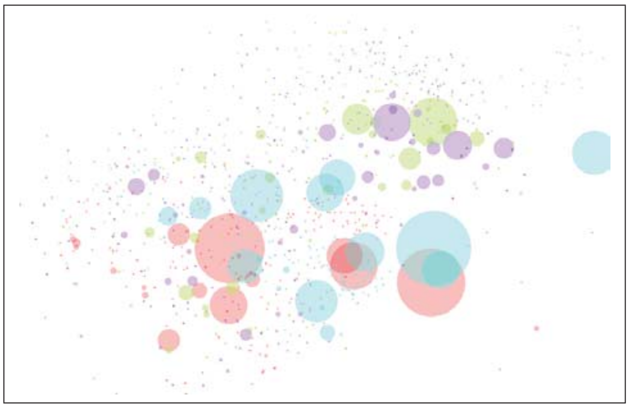

Figure 4: Classical, flat rule model without hierarchy information trained on the vehicle silhouettes dataset showing 222 rule for four classes.

ISBN: 978-989-95079-6-8

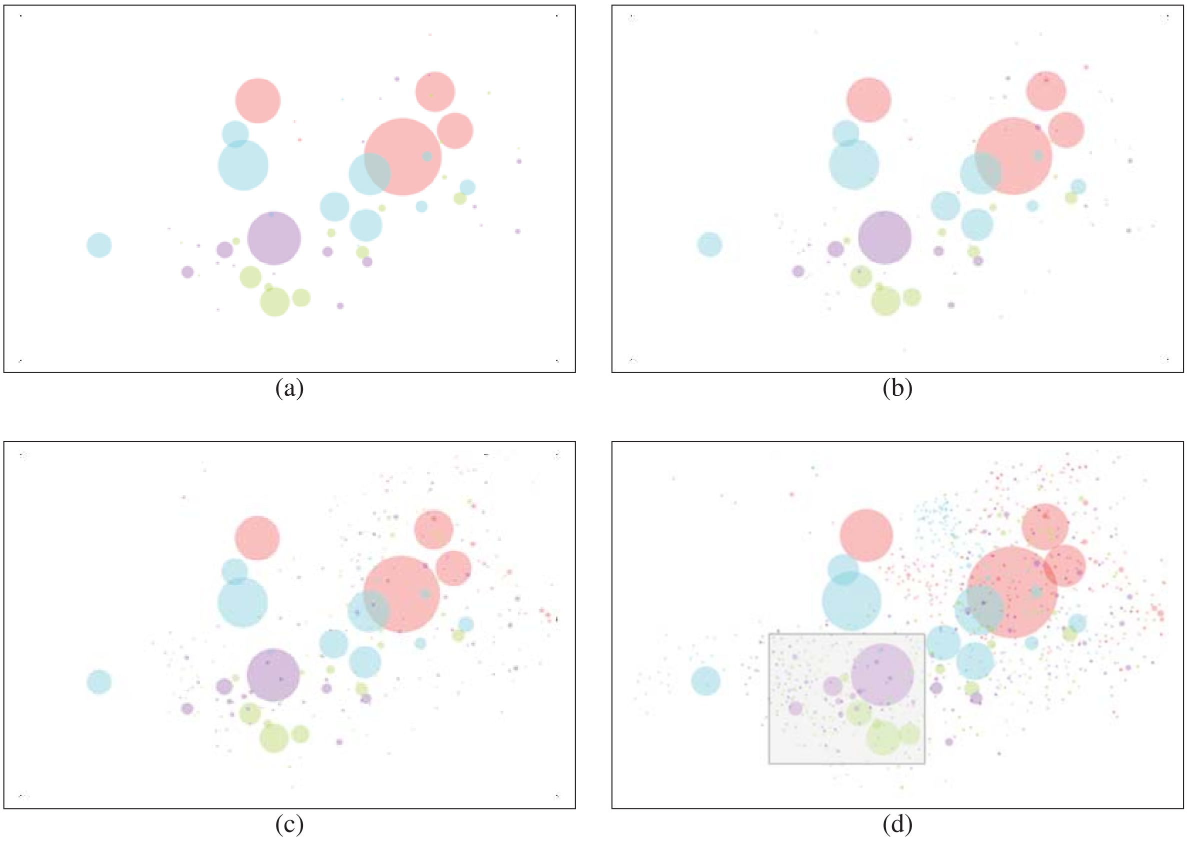

Figure 5: Stepwise rule hierarchy projection on vehicle data starting from top- to bottom-level (a)-(c), and (d) rules of all levels and data points.

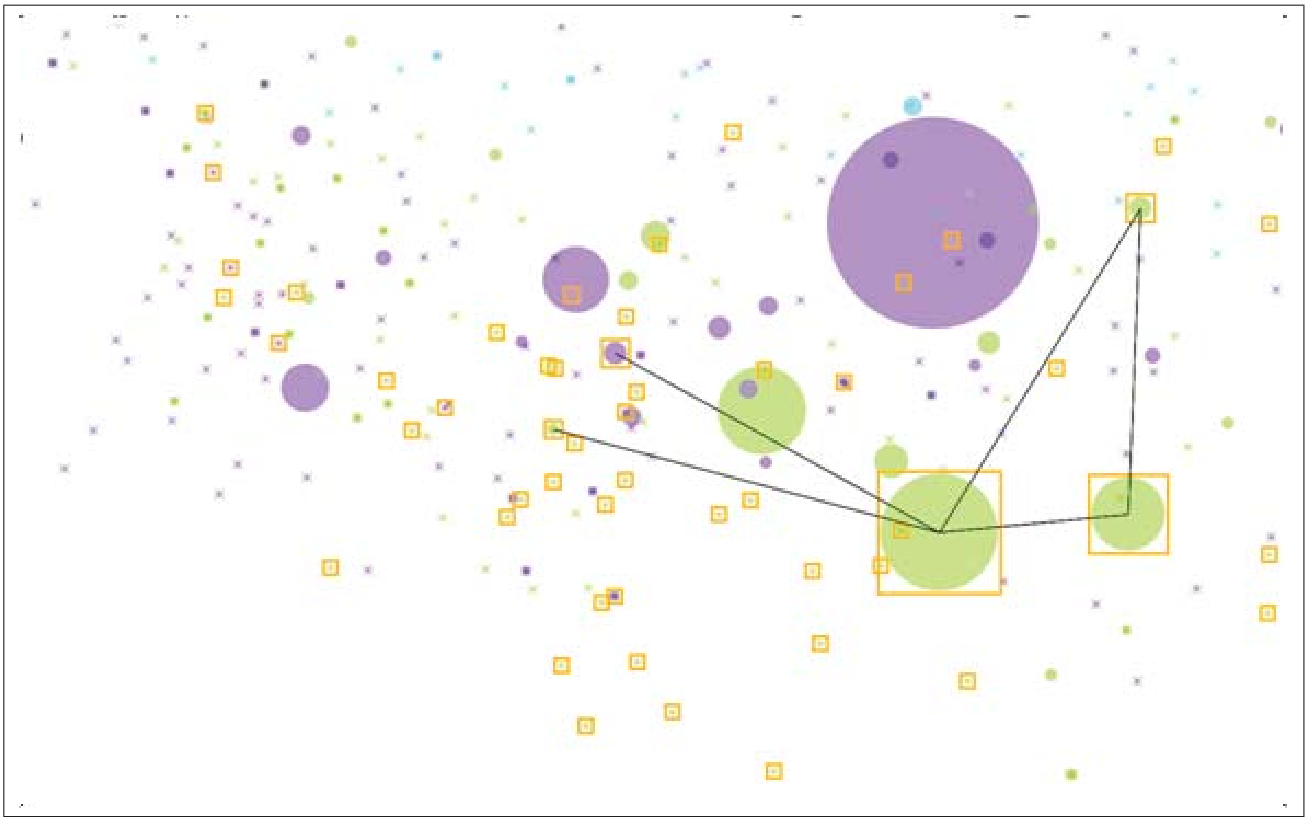

Figure 6: Exploration by zooming and highlighting (orange square) interesting rules and data points in the vehicle hierarchy.

sible since overlapping information is only available between rules of the same single level [1].

The hierarchical rule induction algorithm is applied to generate a hierarchy of rule models in a second test. The three levels of the fuzzy rule hierarchy contain 55 rules in the top, 40 in the middle, and 174 at the bottom-most level. Fig. 5 shows each level together with its parent levels starting from (a) to (c), and (d) rules of all levels and data points. All levels are subsequently projected into the two-dimensional space always with respect to their parent level(s), as well as all data points that are projected relatively to all rule levels. Fig. 6 shows the surrounded area from Fig. 5 (d) enlarged. Selecting one rule highlights all overlapping rules in the levels above along with the data points they cover. All highlighted rules (orange square) that cover the same set of patterns are connected by a line to visually identify overlapping rules between different levels of the hierarchy. The figure shows five connected rules where a violet rule from a lower level completely overlaps with a large, green rule of a higher level. This is typically an indication for outliers and artifacts in the data expressed by smaller, more specific rules at lower levels generated by the rule learning algorithm at first place; whereas rules generated at higher levels of the hierarchy explain more general aspects of the data and are usually covered by larger rules. This visual line-up of rules of different levels allows further exploration by highlighting interesting data instances.

5 Conclusions

Hierarchically structured rules induced by a classical rule learning algorithm lead to a well-defined hierarchy of rules where levels further up explain more general aspects and rule models at lower levels concentrate on artifacts or outliers of the underlying concept. Combining this information with a mapping mechanism to visualize both this type of hierarchy and the data points enables interactive exploration across rule levels by focusing on overlapping regions. On the other hand, it highlights data points covered throughout the rule hierarchy. We demonstrated the explorative power of the proposed projective multi-dimensional scaling method based on the vehicle silhouettes dataset, delivering interesting insights into the underlying concept. Our approach is well suited for projecting other hierarchical structured data and rules driving interactive explorative data analysis.

References

[1] S. Abe, and M.S. Lan. A method for fuzzy rules extraction directly from numerical data and its application to pattern classification. *IEEE Transactions on Fuzzy Systems*. 3:18–28, 1995.

[2] M.R. Berthold. Learning fuzzy models and potential outliers. *In Computational Intelligence in Data Mining.* Springer, Berlin, 111–126, 2000.

[3] M.R. Berthold. Mixed Fuzzy Rule Formation. *International Journal of Approximate Reasoning (IJAR)*. Elsevier, 32:67–84, 2003.

[4] M.R. Berthold, and L.O. Hall. Visualizing Fuzzy Points in Parallel Coordinates. *IEEE Transactions on Fuzzy Systems*. IEEE Press, 11(3):369–374, 2003.

[5] M.R. Berthold, and R. Holve. Visualizing high dimensional fuzzy rules. *Proceedings of NAFIPS*. IEEE Press, 64–68, 2000.

[6] T.F. Cox, and M.A. Cox. Multidimensional Scaling. *Monographs on Statistics and Applied Probability*. Chapman and Hall, 1994.

[7] W. Duch, R. Setiono, and J.M. Zurada. Computational Intelligence Methods for Rule-based Data Understanding. *Proceedings of the IEEE.* IEEE Press, 92(5):771–805, 2004.

[8] Th.R. Gabriel, and M.R. Berthold. Constructing Hierarchical Rule Systems. In: M.R. Berthold, H.-J. Lenz, E. Bradley, R. Kruse, and C. Borgelt (Eds.): *Proc.* 5^{th} *International Symposium on Intelligent Data Analysis (IDA)*. Springer, Berlin, 76–87, 2003.

[9] Th.R. Gabriel, and M.R. Berthold. Influence of fuzzy norms and other heuristics on "Mixed Fuzzy Rule Formation". *International Journal of Approximate Reasoning (IJAR)*. Elsevier, 35:195–202, 2004.

[10] Th.R. Gabriel, A.S. Pintilie, and M.R. Berthold. Exploring Hierarchical Rule Systems in Parallel Coordinates. In: Famili et al. (Eds.): *Proc.* 6^{th} *International Symposium on Intelligent Data Analysis (IDA)*. Springer, Madrid, 97–108, 2005.

[11] Th.R. Gabriel, K. Thiel, and M.R. Berthold. Rule Visualization based on Multi-Dimensional Scaling. *IEEE International Conference on Fuzzy Systems*. IEEE Press, Vancouver, 66–71, 2006.

[12] C.M. Higgins, R.M. Goodman. Learning fuzzy rule-based neural networks for control. *Advances in Neural Information Processing Systems*. California, Morgan Kaufmann, 5:350–357, 1993.

[13] J. Meulman. A distance approach to nonlinear multivariate analysis. DSWO Press, Leiden, The Netherlands, 1986.

[14] D. Michie, D.J. Spiegelhalter, and C.C. Taylor, editors. *Machine Learning, Neural and Statistical Classification*. Ellis Horwood Limited, 1994.

[15] S. Salzberg. A nearest hyperrectangle learning method. *Machine Learning*. 6:251–276, 1991.

[16] F. Rehm, F. Klawonn, and R. Kruse. Visualization of Fuzzy Classifiers. *International Journal of Uncertainty Fuzziness and Knowledge-Based Systems*. World Scientic Publishing Company, 15(5):615–624, 2007.

[17] J.W. Sammon, Jr. A non-linear mapping for data structure analysis. *IEEE Transactions on Computers*. C18(5):401–409, May 1969.

[18] P.K. Simpson. Fuzzy min-max neural networks – part 1: Classification. *IEEE Transactions on Neural Networks*. 3:776–786, 1992.

[19] L.X. Wang, and J.M. Mendel. Generating fuzzy rules by learning from examples. *IEEE Transactions on Systems, Man, and Cybernetics*. 22:1313–1427, 1992.

[20] D. Wettschereck. A hybrid nearest-neighbour and nearest-hyperrectangle learning algorithm. *Proceedings of the European Conference on Machine Learning*. 323–335, 1994.

[1] In some rare cases, neighboring data points of different classes generate rules with overlapping core-regions.