Journal of Universal Computer Science, vol. 11, no. 11 (2005), 1835-1848
submitted: 1/9/05, accepted: 1/10/05, appeared: 28/11/05 © *J.UCS*

Connecting Segments for Visual Data Exploration and Interactive Mining of Decision Rules

Francisco J. Ferrer–Troyano
(Computer Science Dept., Univ. of Seville, Spain
ferrer@lsi.us.es)

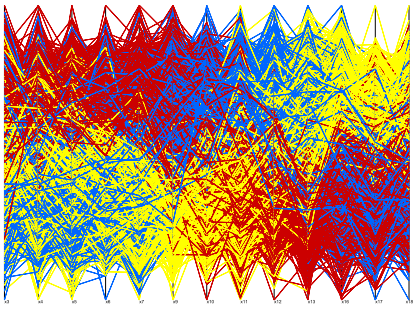
Jesús S. Aguilar–Ruiz
(Computer Science Dept., Univ. Pablo de Olavide, Spain
direscinf@upo.es)

José C. Riquelme
(Computer Science Dept., Univ. of Seville, Spain
riquelme@lsi.us.es)

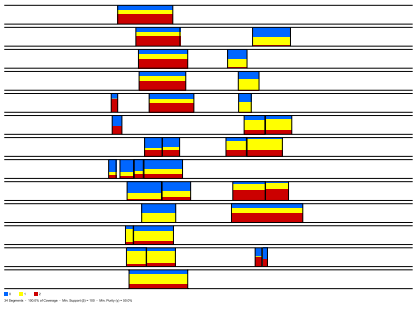
Abstract: Visualization has become an essential support throughout the KDD process in order to extract hidden information from huge amount of data. Visual data exploration techniques provide the user with graphic views or metaphors that represent potential patterns and data relationships. However, an only image does not always convey high–dimensional data properties successfully. From such data sets, visualization techniques have to deal with the curse of dimensionality in a critical way, as the number of examples may be very small with respect to the number of attributes. In this work, we describe a visual exploration technique that automatically extracts relevant attributes and displays their ranges of interest in order to support two data mining tasks: classification and feature selection. Through different metaphors with dynamic properties, the user can re-explore meaningful intervals belonging to the most relevant attributes, building decision rules and increasing the model accuracy interactively.
Key Words: Data Mining, Visual Data Exploration, Connecting Segments
Category: E.1, E.2, H.4

1 Introduction

Visualization techniques provide an important support to extract knowledge from huge amounts of data by incorporating ingenuity, analytic capability, and experience of the user, which makes easier to steer the KDD process. From visual metaphors giving graphic representations of a query or data set, visual data exploration allows the user to achieve an interactive search and identify interesting data relationships, from which new hypotheses and conclusions can be drawn. Such hypotheses can be later verified by learning algorithms. Therefore, visual data exploration ought to facilitate getting an insight into data distribution by means of different detail level views in order to reduce the space complexity and obtain simpler that improve the interpretation of results.



(a) Parallel Coordinates from the most relevant attributes obtained by VETIS.



(b) Connecting segments obtained by VETIS.

Figure 1: *Wave–form* data set (5000 examples, 40 attributes, and 3 class labels).

An important issue in multidimensional data visual exploration is to avoid different entities overlapping on the screen. A graphic entity usually represents a data aggregation given in the form of items, examples, or relationships among attribute values. The reason for this is that, if the values are directly displayed, they usually are a significantly small portion of the entire available data. Otherwise, it is likely that the resulting image does not clearly convey important data properties and the exploration becomes a difficult task. In the case of very–large numerical data sets, the number of different values is higher than the screen resolution, making some visualization approaches have indirectly restricted to data size, with respect either the number of examples or the number of attributes. As an example, Figure 1(a) shows the *Wave–form* data set, displayed using the well–known Parallel Coordinates method [10]. Because of the high dimensionality of this data set, individual examples cannot be clearly seen from this display, also preventing the detection of relevant patterns and attributes.

Since it is not easy to provide clear information about attribute relevance unless the method can automatically extract a relevant subset of them, a more useful approach can be to display as few graphic entities as possible in order to represent as large amount of data as possible. The smaller the number of graphical entities containing higher amount of information, the easier and more meaningful the interpretation of results. Based on this approach, in this paper we describe VETIS (**V**isual **E**xploration **T**hrough **I**nteractive **S**egmentation), a visual exploration technique that indirectly approaches two mining task: classification and feature selection. VETIS extracts and segments the most relevant attributes, displaying those intervals meaningful for the user. From a complex data structure that provides additional information about relationships among attributes and examples, the graphical entities displayed by VETIS have been named connecting segments.

A segment represents the class distribution for a group of examples with consecutive values in a dimension. Each segment can be re-displayed both in Parallel Coordinates and as several segments belonging to new dimensions, giving data views in different exploration levels. In addition, connecting segments can be taken as logic conditions to build decision rules from them in parallel.

In order to show the usefulness of our proposal, in this paper we include quite a few figures obtained from multidimensional UCI data sets [5] that describe by themselves the interactive support to the two above mentioned mining task, traditionally achieved with batch learning algorithms.

This paper is organized as follows. Section 2 outlines the state of the art related with visual data exploration. In Section 3, we describe our approach, putting emphasis on the data structure that supports the method and the algorithm, which is divided in three simple steps. Interactive mining examples with VETIS are shown in Section 4, where graphical outputs are displayed together with realted rules interactively built. Finally, in Section 5, the most important conclusions and future work are summarized.

2 Related Work

According to Keim's taxonomy [12], visual exploration techniques can be classified using three orthogonal criteria:

- The data type to be visualized: *one–dimensional* [15], *two–dimensional* [16], *multidimensional* [1, 13], *text & hypertext* [15], *hierarchies & graphs* [4, 6], and *algorithms & software* [8].
- The data representation: *standard 2D/3D displays* [16], *geometrically transformed displays* [9, 10], *icon–based displays* [7], *dense pixel displays* [13], *stacked displays* [11], and hybrid techniques.
- The user interaction way: *dynamic projection* [3], *interactive filtering* [16], *zooming* [14], *distortion*, and *linking & brushing*.

With respect to the data type, VETIS visualizes multidimensional data sets with numerical attributes. Regarding to the second dimension, our proposal belongs to standard 2D techniques. Each graphic entity in VETIS means a meaningful interval belonging to a relevant attribute. These intervals are displayed as multi–colored bars in which the degree of impurity with respect to the class membership can be easily perceived. According to the third category, VETIS displays involve a dynamic projection in which the user can apply zooming and filtering to detect and validate relevant attributes and potential patterns. Dimensionality reduction has been dealt by different visual approaches [2]. VETIS reduces the dimensionality in an interactive manner so as to find meaningful subdomains according to user measures.

3 Connecting Segments

Within the supervised learning, the problem of classification is generally defined as follows. An input finite data set $\mathcal{T}$ of n training examples is given. Every training example is a pair $e = (\overrightarrow{x}, y)$, where $\overrightarrow{x}$ is a vector of m attribute–values (each of which may be numeric or symbolic), and $y \in \mathcal{Y}$ is a nominal class–value named label. Under the assumption there is an underlying mapping function f so that $y = f(\overrightarrow{x})$, the goal is to obtain a model of $\mathcal{T}$ that approximates f as $\hat{f}$ in order to classify non–labelled test examples, so that $\hat{f}$ maximizes the prediction accuracy.

VETIS approaches the classification of multidimensional data sets with numerical attributes by visual building of decision rules from meaningful intervals belonging to the most relevant attributes. A decision rule is a logic predicate of the form: *if antecedent then label*. The antecedent is a conjunction of conditions Attribute$\models$Values, where $\models$ is an operator that states a relationship between a particular attribute $\mathcal{A}_j$ and values of its domain $\mathcal{D}(\mathcal{A}_j)$. In rule learning, an example $e = (\overrightarrow{x}, y)$ is said covered by a rule r if $\overrightarrow{x}$ fulfills or is described by the conditions belonging to the antecedent of r, whatever the label associated with r is. VETIS allows the user to obtain rules associated with several labels, which are interactively formed from intervals belonging to different attributes. For every meaningful interval is displayed the distribution of labels within it and the relationship with other intervals. Thus, the elemental unit of graphic information in VETIS is called connecting segment, described next.

Definition 1 (Connecting Segment) *A connecting segment $\mathcal{S}$ associated with an attribute $\mathcal{A}_j$ is a data structure consisting of three elements ($\mathcal{I}$, $\mathcal{H}$, $\mathcal{IH}$):*

- ***Interval**: $\mathcal{I} = [l, u)$ is a left–closed, right–open interval in $\mathbb{R}$.*
- ***Histogram**: $\mathcal{H} = \{\mathcal{H}_1, \ldots, \mathcal{H}_z\}$ is a histogram with the number of examples for each label in $\mathcal{Y} = \{y_1; \ldots; y_z\}$ that are covered by $\mathcal{I}$. An example e_i is covered by an interval $\mathcal{I}$ associated with the attribute $\mathcal{A}_j$ if the attribute–value (x_{ij}) belongs to the interval $\mathcal{I}$.*
- ***Overlaps**: $\mathcal{IH}$ is a set of m-1 elements, one per each attribute $\mathcal{A}_k \neq \mathcal{A}_j$. Each element of this set is composed by a set of pairs $(k'; \mathcal{H}_{k'})$, related to segments for other attribute $\mathcal{A}_k$ containing examples covered by $\mathcal{I}$. The element k' is the index of a segment $\mathcal{S}_{k'}$, and $\mathcal{H}_{k'}$ is the histogram of class labels for examples in the intersection $\mathcal{I} \cap \mathcal{I}_{k'}$.*

The purpose of this structure is to compute the minimal set of segments efficiently from which data label distribution can be clearly visualized. This process is illustrated in Algorithm 1 and divided into three steps:

Algorithm 1 VETIS - computing the minimal set of segments

INPUT $\mathcal{T}$: Set of n examples and m atributes; δ, γ: integer
OUTPUT MS: Minimal Set of Segments
begin
Build the initial set of segments IS [Step–1]
Join consecutive initial segments JS [Step–2]
Build the minimal set of segments MS [Step–3]
end

1. First, an initial set of segments is computed (step 1);
2. Second, the segments are analyzed in order to refine them by means of joins that preserve a measure of impurity γ (step 2);
3. Third, the minimal set of segments is generated according to γ together with a measure of coverage β (step 3). Every set can be displayed in order to get an insight into the potential complexity of the final segments.

3.1 Initializing segments

This first phase builds m initial sets IS_j, one per attribute $\mathcal{A}_j$. Each set IS_j is formed by α_j connecting segments and provide the user with insight about the label distribution of input data. The different values of α are calculated by means of projections, i.e., the number of intervals that contain examples for an only class label. Every two adjacent intervals have different class. At least, there will be z initial segments per attribute, where z is the number of different labels ($\mathcal{Y} = \{y_1, \ldots, y_z\}$). This situation is ideal, and it happens when it is possible to obtain z segments, each one of them containing all the examples of that class. In the worst case, there will be as much segments as n, with n being the number of examples. In that case, each segment contains only one example.

The initial sets of segments are built by one only scan, previously generating α empty segments for each attribute with $\mathcal{H}_p = 0$ ($p \in \{1; \ldots; z\}$) and $\mathcal{IH}_k = 0$. Then every example $e_i = (x_i; y_i)$ updates the class labels histogram of the segment $\mathcal{S}$ that covers x_i (increasing by one the $\mathcal{H}_p$ associated with the label y_i), and the relationships $\mathcal{IH}_k$ among such updated segments.

The complexity of this step is mainly determined by the sort algorithm and the method to generate the cutpoints. The latter one takes linear time, therefore the overall complexity is $\Theta(m^2 lg(m))$. The simplest way to obtain the cutpoints consists in fixing a new interval every time a change of label is found. Consecutive values associated with the same label will compose a common segment whereas a value for which there are several examples of different labels will most likely generate a segment where $\mathcal{I} = l = u$.

Algorithm 2 VETIS - Step–1

INPUT $\mathcal{T}$: Set of n examples and m atributes
OUTPUT IS: Initial Set of Segments
begin
  for all attribute $\mathcal{A}_j$ in $\mathcal{T}$ **do**
    Sort attribute values
    for all change of label in $\mathcal{A}_j$ **do**
      Set a new interval
      Calculate histograms for each class and each segment
end

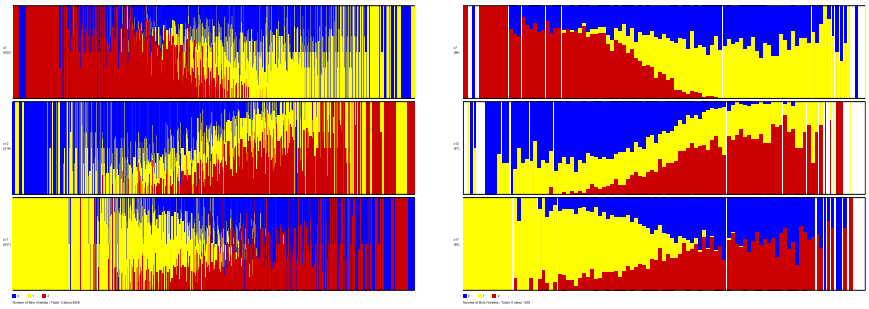
The fist step of the overall process is shown in Algorithm 2, whose purpose is to initialize the data structure that supports the final display. The additional cost required to compute the relationships among segments is not expensive since the index k of a segment $\mathcal{S}$ associated with the attribute–value x_{ij} can be calculated directly with the following expression:

$$k = \lfloor norm(x_{ij}) \times \alpha \rfloor;\ norm(x_{ij}) = \frac{x_{ij} - MIN_j}{MAX_j - MIN_j} \tag{1}$$

where MIN_j and MAX_j are the lower and upper bounds of the attribute range $\mathcal{D}(\mathcal{A}_j)$, and α_j is the number of segments for attribute $\mathcal{A}_j$. Furthermore, VETIS can incrementally reduce the number of segments by joining consecutive segments with equal distribution. Let $\mathcal{S}_a$ and $\mathcal{S}_b$ be two consecutive segments with associated histograms $\mathcal{H}^a$ and $\mathcal{H}^b$, respectively. They are grouped if:

$$\frac{\mid \mathcal{H}_p^a \mid}{support(\mathcal{S}_a)} = \frac{\mid \mathcal{H}_p^b \mid}{support(\mathcal{S}_b)};\ \forall p \in \{1, \ldots, z\}$$

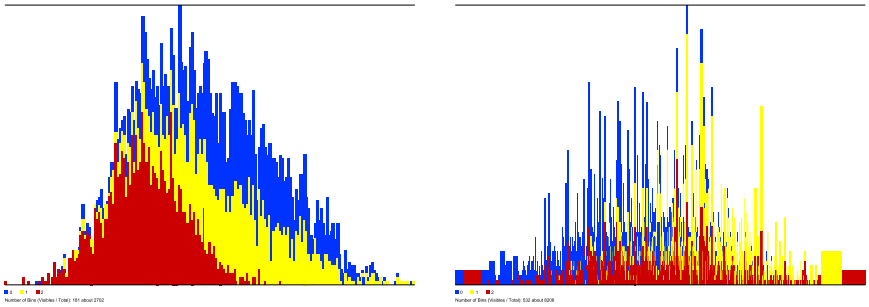
Alternatively, initial segments can be computed using the same α–value in all the attributes (Figure 2). By this option, α equal–width empty intervals are generated for every attribute so that histograms are incrementally completed according to Equation 1. In addition, segments can be displayed in the form of both regular bar charts and equal–width bar charts (Figure 3). As pointed out in [13], the advantage of equal–height bar charts is a better use of the available screen space, but this comes at the disadvantage that the presented items are harder to compare. Although VETIS displays seem very similar to Keim & Hao's Hierarchical Pixel Bar Charts [13], our approach does not belong to pixel–based techniques since the main goal is not to represent input data directly. Contrary, VETIS is based on data aggregation in order to provide interactive rule mining from different graphic entities. VETIS provides displays of the eight options in order to get an insight into the potential complexity of the final minimal set (see Figures 2 and 3).



(a) Different α–value option ($\alpha_7 = 652$, $\alpha_{12} = 516$, and $\alpha_{17} = 531$).

(b) Equal α–value option ($\alpha = 100$).

Figure 2: *Wave–form* data set. Initial segments in three attributes (x7, x15, and x16) displayed as equal–width bars.



(a) Attribute x5. Fixed α–value equal to 200 (181 segments).

(b) Attribute x10. Dynamic α–value (532 segments).

Figure 3: *Wave–form* data set: initial segments in attributes x5 and x10 displayed separately as regular bars.

3.2 Joining segments

In the second phase, previous initial segments are refined in order to obtain m smaller sets JS_j, one per each initial set IS_j ($j \in \{1, \ldots, m\}$). The new segments are obtained by union of consecutive initial segments from a measure of impurity biassing in favour of the attributes with least number of segments and smaller intersection among them. Some definitions related to this step are provided next.

Definition 2 (Pure Segment) *A pure segment $\mathcal{S}$ represents an interval $\mathcal{I}$ of the j^{th} attribute $\mathcal{A}_j$ for which all the examples are associated with the same class label:*

$$\nexists\, e_i,\ e_{i'} \in \mathcal{T} \cdot x_{ij} \in I \wedge x_{i'j} \in I \wedge y_i \neq y_{i'}$$

Algorithm 3 VETIS - Step–2

INPUT IS: Set of Initial Segments; δ, γ: integer
OUTPUT JS: Set of Joint Segments
begin
  for all attribute $\mathcal{A}_j$ **do**
    repeat
      $\mathcal{S}_{best} \leftarrow \emptyset$
      for all pair of consecutive impure segments $(\mathcal{S}_a, \mathcal{S}_b) \in IS_j$ **do**
        $\mathcal{S}' \leftarrow \mathcal{S}_a \cup \mathcal{S}_b$
        if purity($\mathcal{S}'$) $\geq \delta$ **and** support($\mathcal{S}'$)>support($\mathcal{S}_{best}$) **then**
          $\mathcal{S}_{best} \leftarrow \mathcal{S}'$
      if $\mathcal{S}_{best} \neq \emptyset$ **then**
        Replace $\mathcal{S}_a$ and $\mathcal{S}_b$ with $\mathcal{S}_{best}$
    until $\mathcal{S}_{best} = \emptyset$
    for all *segment* $\mathcal{S}_j$ in IS_j **do**
      if support($\mathcal{S}_j$)$\geq \gamma$ **and** purity($\mathcal{S}_j$)$\geq \delta$ **then**
        $JS_j \leftarrow JS_j \cup \{\mathcal{S}_j\}$
end

Definition 3 (Impure Segment) *An impure segment $\mathcal{S}$ represents an interval $\mathcal{I}$ of the j^{th} attribute $\mathcal{A}_j$ for which there are examples associated with different class labels:*

$$\exists\ e_i,\ e_{i'} \in \mathcal{T} \cdot x_{ij} \in I \wedge x_{i'j} \in I \wedge y_i \neq y_{i'}$$

Definition 4 (Support) *The support of a segment $\mathcal{S}$ is the number of examples covered by $\mathcal{S}$:*

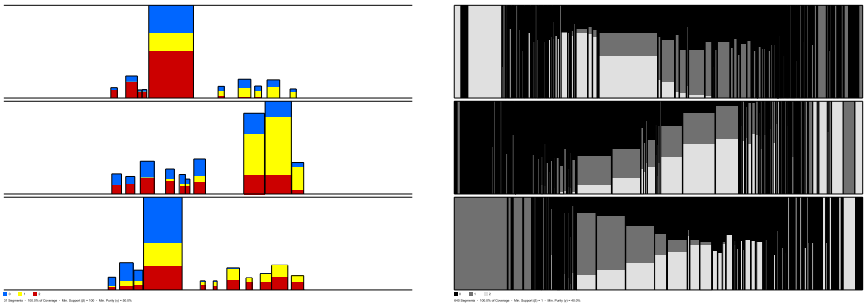
$$support(\mathcal{S}) = \sum_{p=1}^{z} \mid \mathcal{H}_p \mid$$

Definition 5 (Purity) *The purity of a segment $\mathcal{S}$ is the percentage of examples covered by $\mathcal{S}$ with a majority label with respect to its coverage:*

$$purity(\mathcal{S}) = \max_{p=1}^{z} \frac{\mid \mathcal{H}_p \mid}{support(\mathcal{S})}$$

Definition 6 (Minimum Support δ) *The minimum support δ is the lowest support that a segment must surpass to belong to the* Minimal Set of Connecting Segments *(MS).*

Definition 7 (Minimum Purity γ) *The minimal purity γ is the lowest percentage of examples with a majority label with respect to the number of examples covered by an impure segment in order to belong to MS.*



(a) Joint segments in regular bars using $\beta = 100$ and $\gamma = 0.5$.

(b) Joint segments in grey scale using $\beta = 1$ and $\gamma = 0.4$.

Figure 4: *Wave–form* data set. Joint segments in x7, x15, and x16.

The JS sets are built by an iterative procedure (see Alg. 3). For each attribute $\mathcal{A}_j$, VETIS searches the IS_j for consecutive impure segments whose union is possible and whose resulting support is the highest. Two consecutive impure segments can be joined if the resulting purity is greater than or equal to the minimum purity γ. The user can set both parameters δ and γ initially. The parameter δ controls indirectly the size of the segment (number of examples included in the segment). The parameter γ deals with the distribution of classes within segments. By default, δ is set to 1, because user can be interested in any valid segment, and γ to 95% as pure segments are preferred. If the parameters δ and γ exceed the coverage and purity values, respectively, for a specific segment that has been recently joined, then both segments can be definitely joined.

3.3 Building the minimal set

In the last phase, the goal is to find the least number of segments from which to visualize the label distribution, transforming thousands of examples with dozens of attributes into few intervals that can be clearly separated in the display. An iterative procedure adds joined segments from the JS to the MS (see Alg. 4).

In each iteration, only a new segment is included in the MS: the one with the largest number of examples that are not yet covered by other segments already included in the MS set. Thus, the first segment to be included will be the one with the highest support. The procedure ends when either all the examples have been covered or there is no segment that covers examples non–covered by the MS set. The number of new examples Δ associated with an attribute $\mathcal{A}_j$ that a segment $\mathcal{S}_j \in JS_j$ can provide for the MS set is computed by the intersection among $\mathcal{IH}_j$ and all the histograms $\mathcal{H}'$ associated with the segments $\mathcal{S}'$ already included in the MS set. $\mathcal{S}'$ may not be necessarily associated with $\mathcal{A}_j$.

Algorithm 4 VETIS - Step–3

INPUT JS: Set of Joint Segments
OUTPUT MS: Minimal Set of Segments

```
covered← 0
repeat
  S_best ← ∅
  Δ_best ← 0
  for all attribute A_j do
    for all segment S ∈ JS_j do
      ∂ =| ⋂(IH, H') |; ∀ S' ∈ MS
      Δ =support(S)−∂
      if Δ > Δ_best then
        S_best ← S_jk
        Δ_best ← Δ
      else if Δ = 0 > then
        JS_j ← JS_j \ {S}
  if Δ_best > 0 then
    MS ← MS ∪ {S_best}
    covered←covered+Δ
until (Δ_best = 0) or (covered= n)
end
```

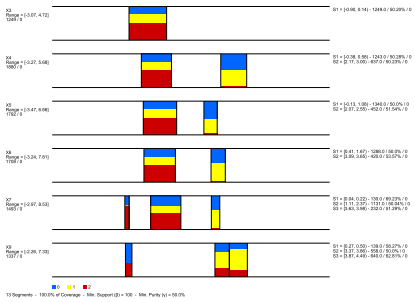
The number of examples in the intersection is computed according to Equation 2:

$$\Delta = support(\mathcal{S}_j) - | \bigcap_{\forall\ \mathcal{S}' \in\ MS} (\mathcal{IH}', \mathcal{H}_j) | \qquad (2)$$

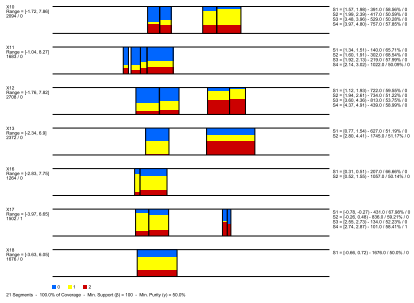
In each new iteration, the number of examples uncovered by non–included segments may change with respect to the previous iteration, so that every segment is re–visited again. If a segment $\mathcal{S} \in JS_j$ does not contain examples uncovered by the MS set, then it is removed from the JS_j set and the next iteration will have lower computational cost. The number of examples Δ are calculated for each segment and the one with greater value is selected to be inserted into MS and removed from JS sets.

3.4 Displaying segments

When the overall procedure ends, the MS set is displayed according to the same graphic representation used in the previous phases (see Fig. 5). For each attribute with at least one connecting segment, a horizontal attribute–bar shows its segments in increasing order of values, from left to right. Every attribute–bar is equal in size, both width and height.



(a) Segments in x3, x4, x5, x6, x7, and x9.



(b) Segments in x10, x11, x12, x13, x16, x17, and x18.

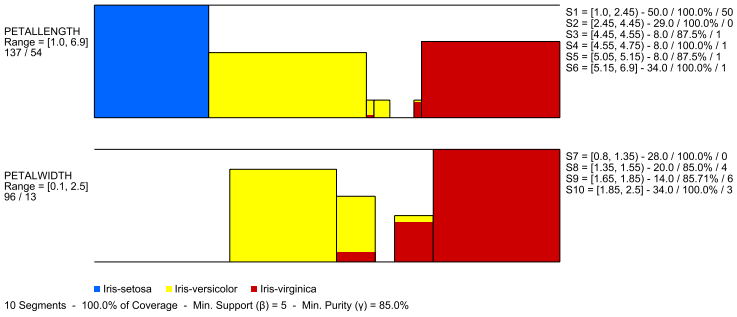
Figure 5: *Wave–form* data set. Minimal set of segments using $\beta = 100$ and $\gamma = 0.5$.

To the left of each bar, VETIS optionally displays the total number of examples covered by all the segments belonging to the related attribute along with the number of exclusive examples that such segments provide. To the right of each bar, the interval of each segment can also be optionally shown together with the number of covered examples, both shared and exclusive examples.

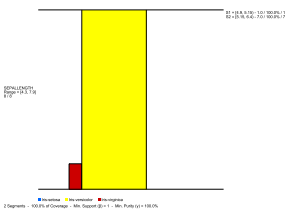
Final segments are displayed as colored bars. The color represents the class label and can be previously selected by the user. Pure segments are displayed with one color and impure segments are displayed with different colors, one per class label with a value greater than 0 in the related histogram. Inside impure segments, every color fills an area on the screen proportional to the number of examples with the label associated to such a color in the respective segment.

4 Interactive Mining

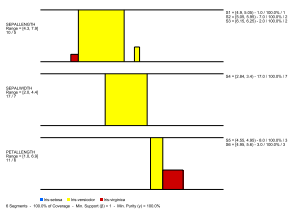
VETIS provides interactive properties so that the user can keep on exploring examples belonging to impure segments until finding a meaningful visual description by means of both Parallel Coordinates technique and new sets over different attributes whose segments have higher purity. In this way, re-exploring impure segments gives a greater insight of data and allows to steer the mining process in order to find, group and validate decision rules on demand. When a segment is re-explored, the subsequent segments represent sub–domains satisfying a higher accuracy. Each new exploration level gives a more detailed description of one condition belonging to the antecedent of a decision rule, since when a segment is re-explored, the subset defined by that condition is visualized. The process is completely interactive, reducing the support and increasing the accuracy every time.



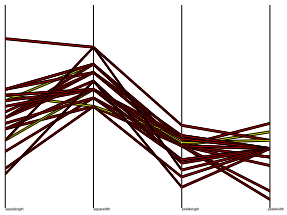
(a) Level 1: minimal set of segments using $\beta = 5$ and $\gamma = 0.85$.



(b) Level 2: minimal set for the examples covered by $\mathcal{S}_3$.



(c) Level 2: minimal set for the examples covered by $\mathcal{S}_3$ and $\mathcal{S}_8$.



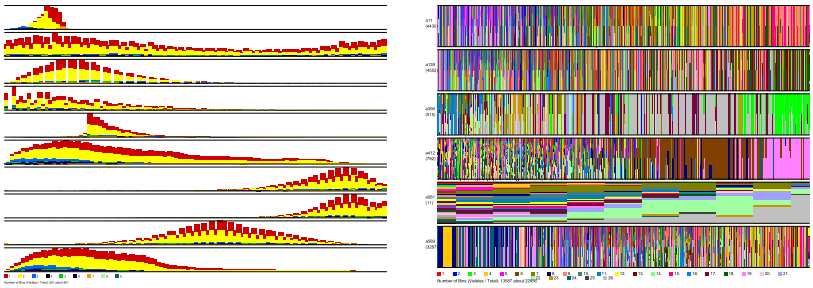
(d) Level 2: examples covered by $\mathcal{S}_5$ and $\mathcal{S}_9$ using Parallel Coordinates.

Figure 6: *Iris* data set. Interactive mining.

Figure 6 shows an example of interactive mining from the *Iris* data set. The resulting segments of the first exploration level give five decision rules:

$$\begin{gathered}
\textit{petal–length} \in [1.0, 2.45) \Rightarrow 100\%\ \textit{setosa} \\
\textit{petal–length} \in [5.15, 6.9] \textbf{ or } \textit{petal–width} \in [1.85, 2.5] \Rightarrow 100\%\ \textit{virginica} \\
\textit{petal–length} \in \{[2.45, 4.45); [4.55, 4.75)\} \textbf{ or } \textit{petal–width} \in [0.8, 1.35) \Rightarrow 100\%\ \textit{versicolor} \\
\textit{petal–length} \in [5.05, 5.15) \textbf{ or } \textit{petal–width} \in [1.65, 1.85) \Rightarrow \textit{virginica} \textbf{ or } \textit{versicolor}\ (86/14) \\
\textit{petal–length} \in [4.45, 4.55) \textbf{ or } \textit{petal–width} \in [1.35, 1.55) \Rightarrow \textit{versicolor} \textbf{ or } \textit{virginica}\ (86/14)
\end{gathered}$$

Pure segments associated with the same label build an only decision rule whereas impure segments are grouped according to the majority class. The latter ones can be re-explored to increase the model accuracy by clicking on them and setting new values for δ and γ. VETIS also provides textual information about the label distribution of the examples covered by each segment and the relationships among them. Thus, the purity of $\mathcal{S}_8$ is 85% due to it covers 17 examples associated with the label *versicolor* and 3 examples with class *virginica*. In addition, 4 examples are only covered by $\mathcal{S}_8$, whereas the rest is shared with other segments belonging to different attributes. Figure 6(b) shows two pure segments of second level connected to $\mathcal{S}_3$. Similarly, two new pure segments can be found from $\mathcal{S}_8$, so that the last rule above can be divided into two accurate



(a) *Covtype* data set. Initial segments in regular bars using $\alpha = 100$.

(b) *Isolet* data set. Initial segments in equal–width bars using different α–values.

Figure 7: VETIS with high–dimensional and very large data sets.

rules:

$$(petal\text{–}width \in [1.35, 1.55) \textbf{ and } petal\text{–}length \in [3.9, 4.95)) \textbf{ or } (petal\text{–}length \in [4.45, 4.55) \textbf{ and } sepal\text{–}length \in [5.15, 6.4)) \Rightarrow 100\%\ versicolor$$
$$(petal\text{–}width \in [1.35, 1.55) \textbf{ and } petal\text{–}length \in [4.95, 5.6)) \textbf{ or } (petal\text{–}length \in [4.45, 4.55) \textbf{ and } sepal\text{–}length \in [4.9, 5.15)) \Rightarrow 100\%\ virginica$$

Furthermore, VETIS allows to link several impure segments in order to visualize the examples covered by all of them. Figure 6(c) shows an alternative minimal set of second level for all the examples covered by $\mathcal{S}_3$ and $\mathcal{S}_8$. Although the number of graphic entities is higher than in the previous case, this new set results in two simpler rules due to all the attributes are taken:

$$sepal\text{–}width \in \{[5.05, 5.95); [6.15, 6.25]\} \textbf{ or } (sepal\text{–}width \in [2.84, 3.4] \textbf{ or } petal\text{–}length \in [4.55, 495)) \Rightarrow 100\%\ versicolor$$
$$sepal\text{–}length \in [4.9, 5.05) \textbf{ or } petal\text{–}length \in [4.95, 5.6] \Rightarrow 100\%\ virginica$$

Similarly, Figure 6(d) shows a second exploration level for 22 examples covered by the segments $\mathcal{S}_5$ and $\mathcal{S}_9$ using Parallel Coordinates. Although there are only three examples with label *versicolor*, one of them is hardly traced. On the other hand, Figure 7 shows the initial sets obtained from two UCI very large data sets: *Covtype* (581012 examples, 54 attributes, and 7 class labels) and *Isolet* (7797 examples, 617 attributes, and 26 class labels). Because of the high degree of impure segments, both displays reflect the potential difficulty to extract an accurate model by means of a classification algorithm.

5 Conclusions and Future Work

VETIS means an approach for visualization of multidimensional data sets with numerical attributes based on feature selection via segmentation. In addition, VETIS allows interactive mining of flexible decision rules through successive exploration levels in which the user decides the support and the purity a segment must fulfil to be taken as condition belonging to the antecedent of a rule.

Results are very interesting as the tool is versatile and allows to build a model with the accuracy level needed. We are currently improving VETIS by using dense pixel approaches to display segments with variable color intensity in such a way support and purity can be easily perceived. On the other hand, several similarity measures are being addressed in order to arrange the segments on the display with respect to the number of shared examples, making easier to understand the graphical information and identify relevant attributes.

References

1. D. F. Andrews. Plots of high–dimensional data. *Biometrics*, 29:125–136, 1972.
2. M. Ankerst, S. Berchtold, and D.A. Keim. Similarity clustering of dimensions for an enhanced visualization of multidimensional data. In *InfoVis'98*, pages 52–60, 1998.
3. D. Asimov. Grand tour: A tool for viewing multidimensional data. *SIAM J. Science and Statistical Computing*, 6:128–143, 1985.
4. G.D. Battista, P. Eades, R. Tamassia, and I.G. Tollis. *Graph Drawing*. Prentice Hall, 1999.
5. C. Blake and E. K. Merz. Uci repository of machine learning databases, 1998.
6. C. Chen. *Information Visualization and Virtual Environments*. Springer-Verlag, 1999.
7. H. Chernoff. The use of faces to represent points in k–dimensional space graphically. *Journal of Am. Statistical Assoc.*, 68:361–368, 1973.
8. M. C. Chuah and S. G. Eick. Managing software with new visual representations. In *IEEE Information Visualization*, pages 30–37, 1995.
9. P. J. Huber. Projection pursuit. *The Annals of Statistics*, 13(2):435–474, 1985.
10. A. Inselberg and B. Dimsdale. Parallel coordinates: A tool for visualizing multidimensional geometry. In *Visualization'90*.
11. B. Johnson and B. Shneiderman. Treemaps: A space-filling approach to the visualization of hierarchical information. In *Visualization'91*, pages 284–291, 1991.
12. D.A. Keim. Information visualization and visual data mining. *IEEE Trans. Visualization and Computer Graphics*, 8(1):1–8, 2002.
13. D.A. Keim and M.C. Hao. Hierarchical pixel bar charts. *IEEE Trans. Visualization and Computer Graphics*, 8(3):255–269, 2002.
14. N. Lopez, M. Kreuseler, and H. Schumann. A scalable framework for information visualization. *IEEE Trans. Visualization and Computer Graphics*, 8(1):39–51, 2002.
15. L. Nowell, S. Havre, B. Hetzler, and P. Whitney. Themeriver: Visualizing thematic changes in large document collections. *IEEE Trans. Visualization and Computer Graphics*, 8(1):9–20, 2002.
16. D. Tang, C. Stolte, and P. Hanrahan. Polaris: A system for query, analysis and visualization of multidimensional relational databases. *IEEE Trans. Visualization and Computer Graphics*, 8(1):52–65, 2002.