# *Eureka!* : A Tool for Interactive Knowledge Discovery

Giuseppe Manco[1], Clara Pizzuti[1], and Domenico Talia[2]

[1] ISI-CNR
c/o DEIS, Università della Calabria
Via P. Bucci, 41C
87036 Rende (CS), Italy
e-mail: {manco,pizzuti}@isi.cs.cnr.it
[2] DEIS, Università della Calabria
Via Bucci, 41C
87036 Rende (CS), Italy
e-mail: talia@deis.unical.it

**Abstract.** In this paper we describe an interactive, visual knowledge discovery tool for analyzing numerical data sets. The tool combines a visual clustering method, to hypothesize meaningful structures in the data, and a classification machine learning algorithm, to validate the hypothesized structures. A two-dimensional representation of the available data allows a user to partition the search space by choosing shape or density according to criteria he deems optimal. A partition can be composed by regions populated according to some arbitrary form, not necessarily spherical. The accuracy of clustering results can be validated by using a decision tree classifier, included in the mining tool.

## 1 Introduction

The production of high-dimensional data sets in different application domains has grown the interest in identifying new patterns in data that might be of value for the holder of such data sets. Knowledge discovery is the process of analyzing data sets to identify interesting, useful and new patterns and trends in data [7]. The knowledge discovery process is a complex task that can involve the use of different data mining techniques. Data mining finds patterns or models that provide summarization of data while loosing the least amount of information. Examples of models comprise clusters, rules, tree structures, and others. The combination of different models in a Knowledge Discovery process may help users in finding what is interesting and significant in large data sets. The Knowledge Discovery is often referred as an interactive and iterative process that involves the following main phases: 1) data preparation and cleaning, 2) hypothesis generation, 3) interpretation and analysis. The hypothesis generation phase, generally, is completely automatic and realized using data mining algorithms based on machine learning and statistics techniques. A different approach aims at exploiting the perceptual and cognitive human abilities, when a visual representation of data is available, to detect the structure of data.

Visual data mining aims at integrating the human in the data exploration process, harnessing his interpretation abilities to large data sets. The basic idea of visual data mining is to present the data in some visual form, allowing the human to get insight into the data, draw conclusions, and directly interact with the data [13]. Visual data mining is especially useful when little is known about the data and the exploration goals are vague. Since the user is directly involved in the exploration process, shifting and adjusting the exploration goals is automatically done if necessary. Visual data mining exploits data visualization to guide the human user in the recognition of patterns and trends hidden in the data. Some interesting visual data mining experiences are described in [4, 2, 6, 17, 15, 14]. When high dimensional data sets are to be mined, visual data mining tools may benefit of the use of dimension reduction techniques that maintain the main features of data.

In this paper we describe a human assisted knowledge discovery tool, named *Eureka*!, that combines a visual clustering method, to hypothesize meaningful structures in the data, and a classification machine learning algorithm, to validate the hypothesized structures. The tool applies the optimal dimensionality reduction method, known as *Singular Value Decomposition (SVD)* [21], to obtain a two-dimensional representation of the available data, and iteratively asks the user to specify a suitable partition of such a representation. The choice of a partition is demanded to the user, thus allowing the identification of clusters of any shape or any density. A partition can provide a separation of dense regions from regions containing sparse data, or it can be composed by regions populated according to some arbitrary polygonal or spherical regions. The accuracy of clustering results can be validated by using a decision tree classifier included in the mining tool. *Eureka*! has been implemented mainly as an extension of the Weka machine learning library [24]. Weka is a Java library defining standard interfaces for data sets loading and preprocessing (e.g., filter definition), mining algorithms and results representation.

The rest of the paper is organized as follows. Section 2 provides a brief introduction of the mathematical technique underlying the clustering tool. In section 3 we describe the interaction metaphor implemented into the system. In particular, section 3.1 covers the cluster generation technique, while section 3.2 is concerned with the cluster validation technique.

## 2 Background: Singular Value Decomposition

$SVD$ is a powerful technique in matrix computation and analysis that has been introduced by Beltrami in 1873 [1]. More recently it has been used in several applications such as solving systems of linear equations, linear regression [21], pattern recognition [5], statistical analysis [12], data compression [16] and matrix approximation [19].

A *singular value decomposition* of an $n \times m$ matrix $X$ is any factorization of the form

$$X = U \times \Lambda \times V^T$$

where $U$ is an $n \times n$ orthogonal matrix, $V$ is an $m \times m$ orthogonal matrix and $\Lambda$ is an $n \times m$ diagonal matrix with $\lambda_{ij} = 0$ if $i \neq j$. It has been shown that there exist matrices $U$ and $V$ such that the diagonal elements of $\Lambda$ are sorted: $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_m$. The diagonal elements $\lambda_i$ are called *singular values* of $X$ and it has been shown that they are the square root of the eigenvalues of the matrix $X^T X$.

The decomposition can equivalently be written as

$$X = \lambda_1 u_1 \times v_1^t + \lambda_2 u_2 \times v_2^t + \ldots + \lambda_m u_m \times v_m^t$$

where $u_i$ and $v_i$ are column vectors of the matrices $U$ and $V$ respectively, $\lambda_i$ are the diagonal elements of $\Lambda$, and it is known as *spectral decomposition* [12]. $SVD$ reveals an important information about the rank of the matrix $X$. In fact, if $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_r \geq \lambda_{r+1} = \ldots = \lambda_m = 0$ then $r$ is the rank of $X$ [8].

Geometrically this factorization defines a rotation of the axis of the vector space defined by $X$ where $V$ gives the directions, $\Lambda$ the strengths of the dimensions and $U \times \Lambda$ the position of the points along the new axis. Intuitively, the $U$ matrix can be viewed as a similarity matrix among the rows of $X$, i.e. the objects of the data set, the $V$ matrix as a similarity matrix among the columns of $X$, i.e. the features that describe an object, the $\Lambda$ matrix gives a measure of how much the data distribution is kept in the new space [11].

In the data mining area, $SVD$ can be used to identify clusters by analyzing the $U$ matrix. By visualizing the matrix $U \times \Lambda$ and considering only the first $d$ dimensions, where $d \leq 3$, we obtain a compressed representation of the $X$ matrix that approximates it at the best. The $d$ kept terms are known as the *principal components* [12].

## 3 *Eureka!* : A tool for interactive knowledge discovery

*Eureka!* , is a semiautomatic tool for interactive knowledge discovery that integrates a visual clustering method, based on the Singular Value Decomposition technique, and a decision tree classifier to validate the clustering results. *Eureka!* has been implemented as an extension of the Weka machine learning library [24] by integrating additional functionalities such as a supervised discretization technique and visual clustering. *Eureka!* has been designed and implemented with the aim of making repeatable the knowledge discovery process on a data set and storing the steps done during the overall process into a repository in order to use it again at a later time. Thus *Eureka!* implements a fixed model of interaction with the user, in which the various steps of the data mining process are represented in a uniform way and executed according to a predefined schema. To this end *Eureka!* generates a hierarchical structure that describes the overall KDD process called *Repository*. The schema that models the KDD process is shown in fig. 1.

Intuitively, an analysis addressing some predefined objectives defines a *business-process*. A given business process is composed by one or more *kdd-process* items.

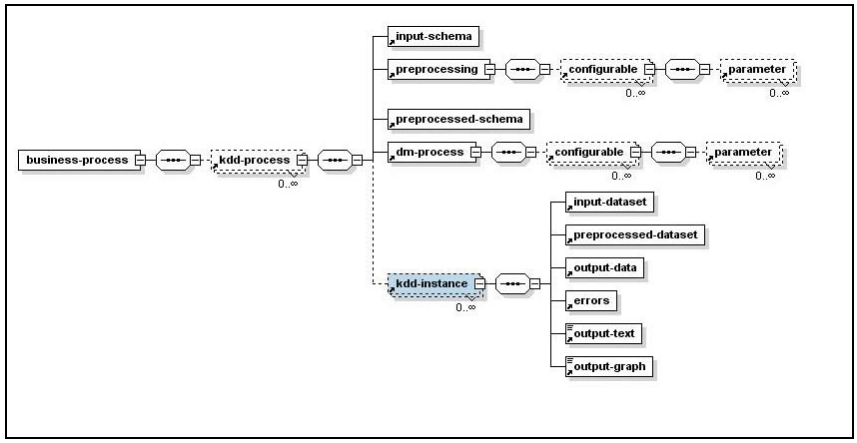


**Fig. 1.** The Interaction Model implemented in *Eureka!* .

Such items have the main objective of describing the meta-schema of each possible instantiation of the analysis: in particular, a *kdd-process* has a given data set (with a given structure described by the *input-schema* item), and it is subject to a given number of *preprocessing* steps, thus providing a *preprocessed-schema* item. The *dm-process* module describes the data mining techniques. Finally, a *kdd-instance* contains one or more possible instantiations of a KDD process. In particular, it contains an input data set conforming to the *input-schema*, the data set resulting from the preprocessing steps described in *preprocessing*, and the resulting patterns obtained from the application of the data mining algorithms.

The methodology employed in *Eureka!* is shown in figure 2. An input data set is transformed into a reduced data set by applying the SVD algorithm and visualized with respect to any two principal components. The user is then asked to choose a portion of the search space he deems interesting. The selected portion is identified as a cluster and the process is repeated on the remaining data until the user judges satisfactory the grouping obtained. At this point each tuple of the data set is labelled with the corresponding class decided by the user and a decision tree inducer can be run to verify the accuracy of the model found. Low misclassification errors should substantiate the detected groups. Thus, if the misclassification error is high, the user can backtrack on his choices and provide an alternative division of the search space, otherwise he can save the process done, and its results. In figure 3 the graphical interface of *Eureka!* is showed. It is composed of three main areas. On the left, the component referred as *Navigator* allows the generation and navigation of the *Repository*. The *Repository* is a hierarchical tree structure that maintains the step sequence done during the overall KDD process. It thus allows the creation and updating of a *business-process*. The bottom part of the interface provides messages about the *kdd-process* execution and the right part shows the current running task.

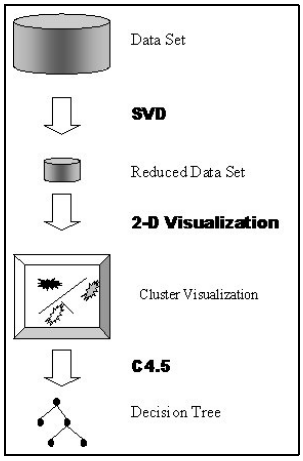


**Fig. 2.** Main steps of the data mining methodology.

We now describe the main features of the system by means of a well-known example: the *image segmentation* database. The data set was taken from the UCI Machine Learning Repository [3] and describes a set of instances drawn randomly from a database of 7 outdoor images. The images where hand-segmented to create a classification for every pixel. The data set consists of 19 numeric attributes, describing the features of a $3 \times 3$ region that the instance represents. From the discussion above, we can detect two main steps in the clustering process: cluster generation and visualization, and cluster interpretation and validation. Let us analyze them in deeper details.

### 3.1 Cluster Generation and Visualization

As mentioned before, *Eureka!* implements an interactive divisive hierarchical clustering algorithm such that at each step clusters can be chosen visually in a two-dimensional space. Such an approach has the advantage of allowing to choose clusters that do not necessarily obey to predefined structural properties, such as density or shape [10, 9]. At each step, a user can choose the cluster according to the criteria that are more likely to be applied. To this end, after the data set has been selected and preprocessed, the clustering task starts by choosing to apply the SVD transformation to the data set, as shown in figure 4 a). Initially, the visualization represents a single node in the cluster tree.

By visualizing the transformed data set, figure 4 b), we can clearly distinguish at least three separate regions. In our visualization, separate regions represent clusters, i.e., elements that can be grouped together. In order to identify clusters, we need to draw the borders of a given region. More precisely, we can choose a region, and separate it from the rest of the space that is represented. *Eureka!* allows the user to separate a region by choosing an appropriate shape, as shown in figure 5 a). Once a region has been selected, we can store such a

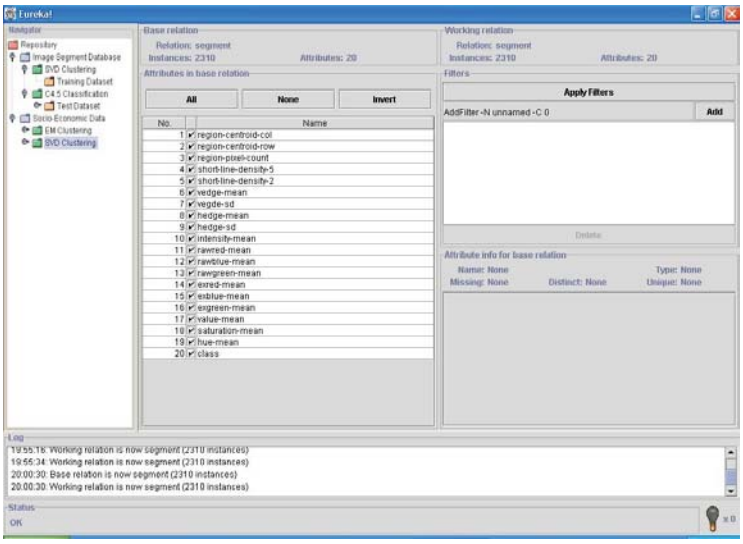


**Fig. 3.** *Eureka!* interface.

selection, thus obtaining a separation of the original space in a cluster tree representing two different groups, as shown in figure 5 b). The right node represents the selected region, and the remaining points are represented by the left node. We can choose any node in the Tree Visualizer Pane, thus allowing the corresponding visualization in the Data Visualizer Pane. In figure 6 the 1358 points of the left node and the 952 points of the right node are visualized.

New nodes can then be recursively split. In particular, the right node shows a clear separation among two different regions, and is worth a further splitting. This is shown in figure 7. An interesting aspect of the tool is the capability of changing axes in the two dimensional representation. By default the mining tool provides a visualization of the first two dimensions (corresponding to the highest eigenvalues of the matrix $\Lambda$). However, by clicking over a given dimension among those shown in the left part of the Data Visualization pane, the user can change such a visualization as needed. A different visualization of the node shown in fig. 6 a), can be obtained by representing the $Y$ axe using the fourth dimension in the SVD representation (fig. 8 a). Different visualizations can help a user in

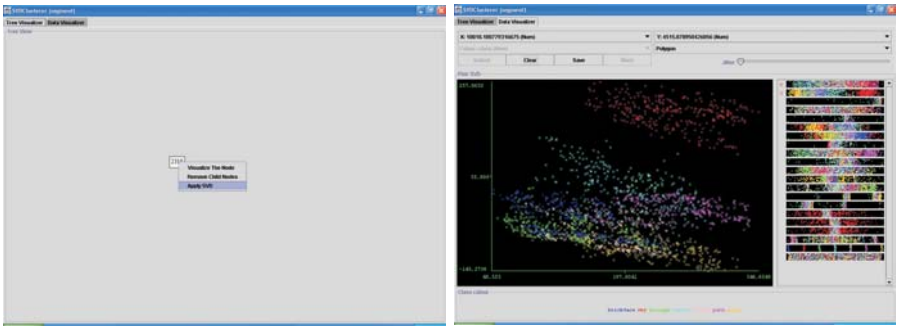

**Fig. 4.** a) *Eureka!* runs the SVD transformation to the overall data set. b) Visualization of the transformed data set with respect to the two principal components.

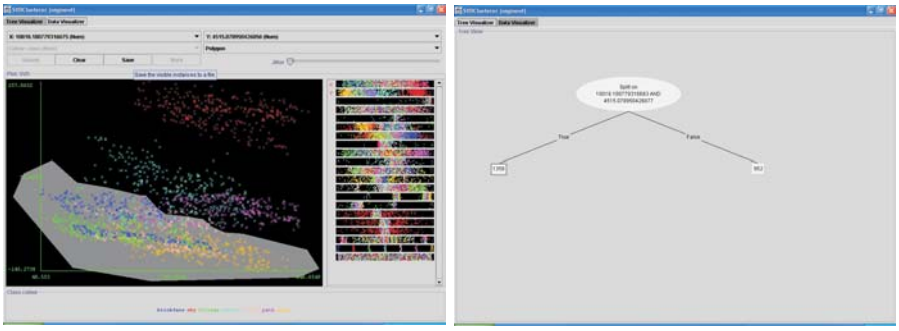

**Fig. 5.** a) Selection of a portion of the data set. b) Cluster tree after the first split.

the cluster identification process. An unsatisfactory partition can be removed by directly acting on the cluster tree. For example, if the analysis of a node does not put in evidence a clear separation of the regions in the given data set partition, we can choose to delete it, as shown in figure 8 b).
Many further choices are available, in order to make separation as accurate as possible. In particular, we can choose non-convex regions, as shown in figure 5 a). The interaction of a given user within the cluster tree is stopped when no further significant splits can be detected. In the *segment* example, we obtained a tree containing 29 nodes and 17 leaves, as shown in fig. 9 a). The leaves of such a tree represent a partition of the data set in 17 groups.

### 3.2 Cluster Interpretation and Validation

A typical problem in clustering algorithm is the problem of assessing the quality of the results. The correctness of a clustering algorithm results has to be validated using appropriate criteria and techniques. Since clustering algorithms define clusters that are not known a priori, irrespective of the clustering meth-

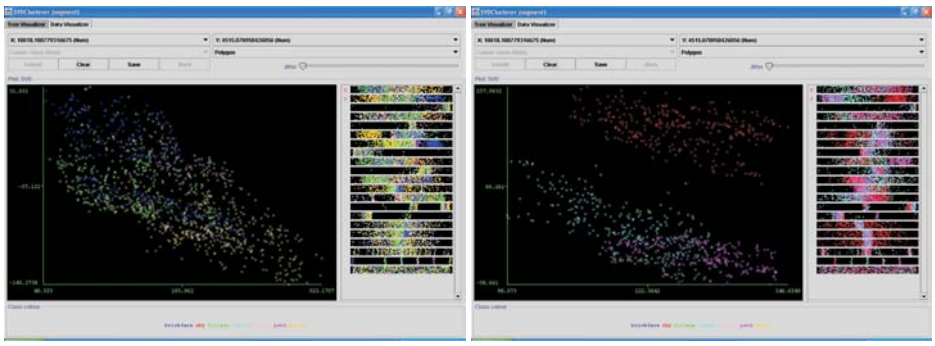

**Fig. 6.** Visualization of the first two selected portions.

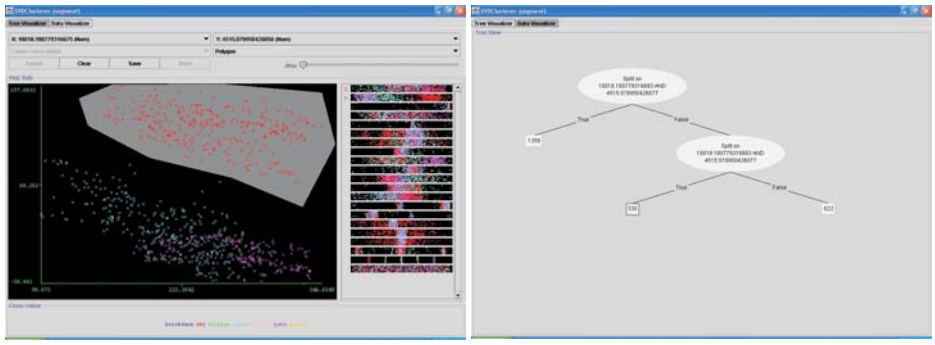

**Fig. 7.** Splitting of the right node.

ods, the final partition of data requires some kind of evaluation [9]. In [23, 9], three main methods are described for assessing the validity of a clustering result:

- *external criteria*, when clustering results are evaluated according to a prespecified structure.
- *internal criteria*, when clustering results are evaluated in terms of the quantities that are computable from the available data (e.g., similarity matrix).
- *relative criteria*, when evaluation takes place in comparison with other clustering schemes.

In particular, when no predefined structure is available, a possibility is to evaluate clustering results using only quantities and features inherent to the data set. For example, one can evaluate the global quality of a clustering scheme by measuring both its *compactness* (i.e., how close the elements of each cluster are to each other) and *separation* (i.e., the difference between two distinct clusters). Clearly, various solutions are possible. For example, we can measure the compactness by looking at the maximal intra-cluster similarity:

$$\max_{\mathcal{C}_i} \sum_{\mathbf{x},\mathbf{y}\in\mathcal{C}_i} d(\mathbf{x},\mathbf{y})$$

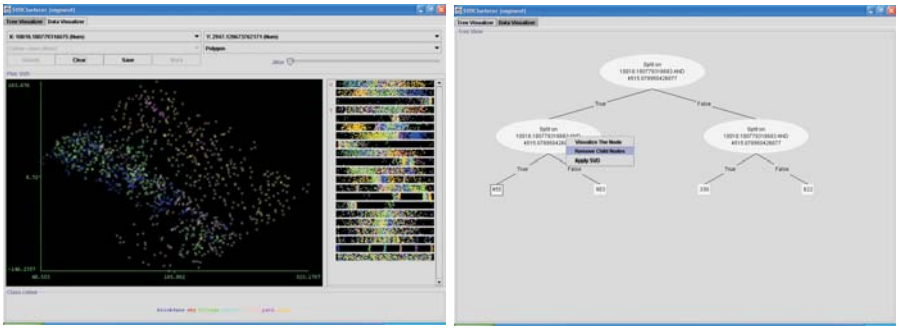

**Fig. 8.** a) Axe Modification. b) Deletion of unsatisfactory partitions.

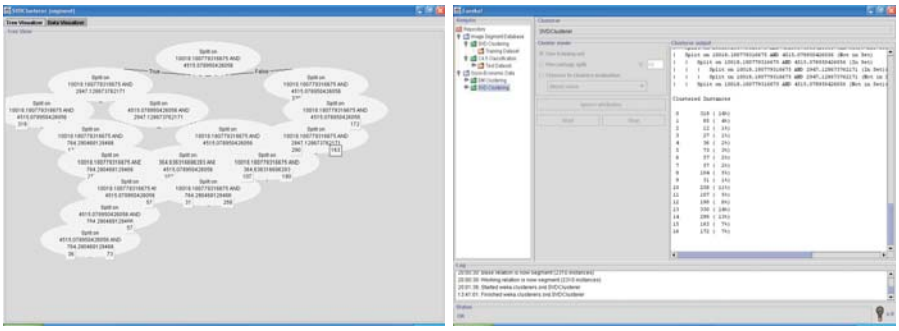

**Fig. 9.** Final Cluster Tree and data set distribution in *Eureka!* .

Many of these methods, however, cannot be used to compare different clustering algorithms. For example, it is not significant to compare algorithm that use different definitions of similarity (or distance). More importantly, often such quality measures are simple statistical indexes that do not describe sufficient properties useful to obtain an interpretation of each cluster.

In *Eureka!* , in order to evaluate the validity of a clustering scheme when no further information is available, we adopted a classification-based internal criterion. Such a criterion is mainly based on the observation that good clusters should be in easily separable regions, and hence a classifier could easily characterize them. Many classification schemes can be applied at this point. In particular, decision-tree classifiers [20] can be well-suited for identifying linearly separable regions. Practically, the criterion for assessing the validity of a cluster is that of building a predictor of the cluster label. A good clustering result should produce a low-error classifier. More importantly, a low-error decision-tree classifier provides a set of rules (directly obtained from the classification tree), that can reveal extremely useful to give an interpretation of each cluster resulting from the application of the clustering algorithm. In order to implement such a validation scheme, we exploited the functionalities available in Weka. Weka cluster interface, in fact, allows to automatically add a cluster label to the data

set under consideration. Starting from this labelled data set, we can set up a new knowledge discovery process, in which we include a decision tree classifier with the aim of predicting the `cluster` attribute. The application of a decision-tree classification algorithm, (e.g., the C4.5 algorithm) ends up with a tree representing the interpretation of the cluster partition, and a set of measures (such as percentage of correctly classified instances, mean error rate, etc.), that assess the validity of the clustering scheme obtained so far.

We used the *Eureka!* tool in several data mining experiments. In particular, recently we used *Eureka!* on a data set of *social-economic data* representing a collection of measurements made in a given number of cities in an Italian district, and concerning social-economic factors such as unemployment rate, amount of companies, amount of agencies, etc.. The resulted knowledge discovery process produced very interesting results [22]. Due to space limit we discuss here the results obtained on the *segment* data set used in the previous sections as a running example. It is useful mainly to show the internal criteria approach. In the visual clustering process, we identified 17 clusters in the data set. Then a decision-tree classifier trained to predict the `cluster` attribute produced a tree with a degree of accuracy of 94% and produced 77 rules describing the features of the discovered clusters. It is interesting to compare such results with the results of a different clustering algorithm. For example, we compared such results with the clustering scheme resulting from the application of the EM algorithm [18] on the same data set. By imposing 7 classes (the optimal number of clusters, obtained via Cross-Validation), we obtain an error rate of 45%. Moreover, suggesting other clustering scheme to EM (e.g., different class numbers) produced a less accurate classification. This test case, as well as the social-economic data analysis, showed that a visual clustering methodology can produce more accurate results compared to that obtained using an oblivious clustering algorithm.

## 4 Conclusions and Future Works

In this paper we described the main features of an interactive knowledge discovery tool, named *Eureka*!, that combines a visual clustering method, to hypothesize meaningful structures in the data, and a classification machine learning algorithm, to validate the hypothesized structures. A two-dimensional representation of the available data allows a user to partition the search space by choosing shape or density according to criteria he estimates optimal. The accuracy of clustering results obtained through the user intervention can be validated by using a decision tree classifier which is a component of the mining tool. We used a simple data set to describe the tool features and how the discovery process is performed using it. Currently we are using *Eureka*! to mine different data sets in several application domains. At the same time, we are working on the tool improvements and extensions like in/out zooming features and automatic region separation suggestions provided to a user by the system at each splitting step.

## References


1. E. Beltrami. Sulle funzioni bilineari [on bilinear functions]. *Giornale di Matematiche ad Uso degli Studenti delle Università*, 11:98–106, 1873.
2. S. Berchtold, H.V. Jagadish, and K.A. Ross. Independence Diagrams: A Technique for Visual Data Mining. In *Proceedings of Fourth Int. Conf. on Knowledge Discovery and Data Mining*, 1998.
3. C.L. Blake and C.J. Merz. UCI repository of machine learning databases, 1998.
4. K.C. Cox, S.G. Eick, G.J. Wills, and R. J. Brachman. Visual Data Mining: Recognizing Telephone Calling Fraud. *Data Mining and Knowledge Discovery*, 1(2):225–231, 1997.
5. R.O. Duda and P.E. Hart. *Pattern Classification and Scene Analysis*. Wiley, New York, 1973.
6. U. Fayyad, G.G. Grinstein, and A. Wierse. *Infomation Visualization in Data Mining and Knowledge Discovery*. Morgan Kaufmann, 2002.
7. U.M. Fayyad, G. Piatesky-Shapiro, and P. Smith. From Data Mining to Knowledge Discovery: an overview. In U. Fayyad et al., editors, *Advances in Knowledge Discovery and Data Mining*, pages 1–34. AAAI/MIT Press, 1996.
8. G.H. Golub and C.F. Van Loan. *Matrix Computation*. The Johns Hopkins University Press, 1989.
9. M. Halkidi, Y. Batistakis, and M. Vazirgiannis. On Clustering Validation Techniques. *Journal of Intelligent Information Systems*. To appear. Available at `http://www.db-net.aueb.gr/mhalk/papers/validity_survey.pdf`.
10. J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufman, 2000.
11. A.K. Jain and R.C. Dubes. *Algorithms for Clustering Data*. Prentice Hall, 1988.
12. I.T. Jolliffe. *Principal Component Analysis*. Springer Verlag, 1986.
13. D.A. Keim and S. Eick. *Proceedings Workshop on Visual Data Mining*. ACM SIGKDD, 2001.
14. D.A. Keim and H.P. Kriegel. Visualization Techniques for Mining Large Databases: A Comparison. *IEEE Transaction on Knowledge and Data Engineering*, 8(6):923–938, 1996.
15. F. Korn et al. Quantifiable Data Mining Using Principal Component Analysis. *VLDB Journal*, 8(3–4):254–266, 2000.
16. F. Korn, H.V. Jagadish, and C. Faloutsos. Efficient Supporting Ad Hoc Queries in Large Datasets of Time Sequences. In *Proceedings of the ACM Sigmod Conf. on Magagment of Data*, 1997.
17. M. Macedo, D. Cook, and T.J. Brown. Visual Data Mining In Atmospheric Science Data. *Data Mining and Knowledge Discovery*, 4(1):68–80, 2000.
18. G.J. MacLahan and T. Krishnan. *The EM Algorithm and Extensions*. Wiley, 1997.
19. W. H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Receips in C: The Art of Computing*. Cambridge University Press, 1992.
20. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.
21. G. Strang. *Linear Algebra and its Applications*. Academic Press, 1980.
22. Telcal Team. Analisi della struttura produttiva ed occupazionale della regione calabria: Risultati. Technical report, Piano Telematico Calabria, 2001. in italian.
23. S. Theodoridis and K. Koutroubas. *Pattern Recognition*. Academic Press, 1999.
24. I. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools with Java Implementation*. Morgan-Kaufman, 1999.