VidaMine: a visual data mining environment

Stephen Kimani\textsuperscript{a}, Stefano Lodi\textsuperscript{b,*}, Tiziana Catarci\textsuperscript{a}, Giuseppe Santucci\textsuperscript{a}, Claudio Sartori\textsuperscript{b}

\textsuperscript{a} Dipartimento di Informatica e Sistemistica, Università di Roma “La Sapienza”, Via Salaria 113, 00198 Roma, Italy
\textsuperscript{b} Dipartimento di Elettronica, Informatica e Sistemistica, Università degli Studi di Bologna, Via Risorgimento 2, 40136 Bologna, Italy

Received 15 August 2002; received in revised form 17 March 2003; accepted 9 June 2003

Abstract

That the already vast and ever-increasing amounts of data still do present formidable challenges to effective and efficient acquisition of knowledge is by no means an exaggeration. The knowledge discovery process entails more than just the application of data mining strategies. There are many other aspects including, but not limited to: planning, data preprocessing, data integration, evaluation and presentation. The human-vision channel is capable of recognizing and understanding data at an instant. Effective visual strategies can be used to tap the outstanding human visual channel in extracting useful information from data. Unlike the case with most research efforts, the exploitation should be employed not just at the beginning or at the end of the knowledge discovery process but across the entire discovery process. In essence, this calls for the development of an effective user/visual component, the development of an overall framework that can support the entire discovery process/all discovery phases, and the strategic placement of the visual component in that framework. Key issues of this component will be the open architecture, allowing extensions and adaptations to specific mining environments, and the precise semantics and syntax, allowing an optimal integration between the presentation and the computation.

© 2003 Elsevier Ltd. All rights reserved.

Keywords: Visual interface; Usability; Data mining

*Corresponding author. Tel.: +39-051-20-93560; fax: +39-051-20-93540.
E-mail addresses: kimani@dis.uniroma1.it (S. Kimani), slodi@deis.unibo.it (S. Lodi), catarci@dis.uniroma1.it (T. Catarci), santucci@dis.uniroma1.it (G. Santucci), csartori@deis.unibo.it (C. Sartori).
1. Introduction

At the heart of virtually any kind of human activity are ever-increasing massive amounts of data. Hardware, databases and network technologies have witnessed tremendous breakthroughs. As a consequence, many individuals and organizations have been able and are still able to acquire, store, share and disseminate vast amounts of data. On the other hand, there have not been corresponding advances in techniques for extracting knowledge from the accumulated vast quantities of data. The knowledge discovery process may be defined as an interactive and iterative non-trivial process that entails: carrying out some initial planning, data integration, selection of target data, data cleaning and pre-processing, data reduction and transformation, selection of suitable data mining techniques to support the discovery process, and evaluation, presentation and interpretation of results from which there exists a subset that may be considered as new “knowledge” [1]. Most of the research efforts in knowledge discovery have concentrated on the development and the optimization of data mining algorithms, with little consideration, if any, of the other knowledge discovery phases and aspects. It is thus of great necessity to consider and model an overall framework that can support the entire knowledge discovery process [2]. In the light of the foregoing, the \(D2I: Integration, Warehousing, and Mining of Heterogeneous Data Sources\) project\(^2\) was initiated. The project aims at defining an overall/comprehensive framework for, among other things, the integration, warehousing, and mining of heterogeneous data sources. In this paper, we concentrate on VidaMine,\(^3\) a system covering the mining part of the \(D2I\) project. VidaMine is a visual data mining system that exploits visual strategies/techniques thereby offering a consistent, uniform, flexible visual interface that is designed based on the goal of allowing or enabling the user not only to process data, but also to steer, guide or direct the entire process of mining knowledge.

Research efforts that concentrate on some specific discovery phase(s) end up with frameworks that are not representative of the entire knowledge discovery process. Besides system extensibility complications, adopting such approaches also normally leads to a non-uniform user interface across different mining tasks within the same system. A framework supporting the entire knowledge discovery process should be open and should integrate all the phases seamlessly. Indeed, since the framework designer does not have prior knowledge of all the components the framework will be expected to support, the framework itself should be extensible to any new components. On the same note, the development of the framework and the components should be separated. VidaMine has been developed in line with the foregoing architectural requirements taken into account.

It is a well-known fact that the human-vision channel has a high bandwidth and that it can surveil a visual field in a parallel manner, while at the same time

---

\(^1\) \(D2I\) is an acronym for Data to Information.

\(^2\) A project supported by the Ministry of Education, University and Scientific Research (Italy). More information about the project may be found at \(http://www.dis.uniroma1.it/~lembo/D2I\).

\(^3\) VidaMine is an acronym for VIstual DAta MINing Environment.
processing the corresponding data even to different levels of detail. It is amazing that the channel can process the data automatically and unconsciously. The human visual system thus enables both recognition and understanding of overwhelming data at an instant [3]. The human visual channel is therefore an outstanding resource that can be exploited in detecting and extracting knowledge from data. Tapping into the human visual system would primarily entail exploiting relevant and effective visual strategies within the user interface. While it is worth acknowledging that many existing mining systems support such strategies, most of them have employed the visual strategies only at the beginning and at the end of the discovery process. Human involvement in the entire mining process is crucial. VidaMine exploits effective visual strategies throughout the entire mining process. In order to effect the human involvement in the entire process, a human-user architectural component is designed and positioned at a strategic place in the aforementioned overall framework. Such an approach constitutes a great step toward according the user a central place in the entire discovery process.

While it is crucial that a visual data mining system involve the user in the entire mining process, we considered it also important to involve the users in the user interface design process. Usability studies provide the recourse for realizing such a goal. VidaMine is a realization of users-to-be actually guiding the interface design process. Usability tests evaluate the extent to which the system can be used by specified users to realize specified goals with effectiveness, efficiency and satisfaction in a specified environment [4]. Most visual data mining systems have given little, if any, consideration to usability evaluation in the development life-cycle.

Most visual data mining systems do not give a precise definition of the syntax of the visual interface and the corresponding semantics. Among other benefits, such a definition would facilitate data exchange, capturing semantics and process automation.

We should also observe that our system is general purpose i.e. not tailored to a specific data mining environment. It is well known that many general data mining systems have been unsuccessful, nevertheless, we believe that our system can easily fit in particular applications for three main reasons: its open architecture, its precise semantics and its friendly user interface.

The key features of VidaMine include:

- an open architecture for the system, which is split into a user layer and a data mining layer;
- a set of infrastructural services allowing interaction between the various software components and providing a clean interface for forthcoming system extensions;
- a consistent, uniform, flexible visual interface based on the goal of supporting the user across the entire data mining process;
- a real user-centered user interface design, equipped with usability studies. The usability studies have been progressively carried out on the prototype versions of the system; these studies involved expert users, such as data miners, statisticians and data analysts, and casual users, such as managers;
the uniform presentation of different mining techniques; this was necessary in order to reach the desired level of integration between system components and, to the best of our knowledge, this is the first attempt to present a uniform framework for the clustering algorithms and to offer a user interface specially for metaquerying;

- a careful definition of visual syntax and formal semantics; each kind of data mining algorithm considered has been carefully analyzed in order to point out the precise meaning of each user choice.

All the unique characteristics of the system are discussed in the rest of the paper. However, in this paper we mainly focus on the system’s visual aspects that accord support to the user in the entire mining process. More details on the visual syntax and formal semantics of VidaMine can be found in [5]. The organization of the rest of the paper is as follows: Section 2 presents related research work. Section 3 sheds light on the framework of VidaMine. Section 4 concentrates on the visual interface of the system. Usability studies are presented in Section 5. Future research direction and conclusions are presented in Section 6.

2. Related work

In this section is a discussion on some systems/tools that offer a reasonably great and diverse number of data mining and visualization functionalities.

Clementine [6] is a product by Integral Solutions Ltd (ISL). SPSS purchased ISL on December 31, 1998. The product supports quite a number of mining techniques including: clustering, association rules, sequential patterns, factor analysis, and neural networks. Its visual interface reveals much about a data mining task by illustrating the flow of control and data. Therefore the user is better positioned to understand and follow the mining process. Users construct a map of their data mining project/model, called a “stream”, by selecting icons, called “nodes”, that represent steps in the data mining process. However, users would need to learn and think in terms of “streams” and “nodes”. It should be pointed out that Clementine does allow users to adjust/refine their “streams” and rerun the system on the refined model. It would be interesting for the company to have evaluation studies carried out and reported for the express purpose of assessing usability of Clementine.

Enterprise Miner [7] is a product by the SAS Institute. The product provides diverse data mining algorithms such as: decision trees, neural networks, regression, radial basis functions, and clustering. The product offers extensive parameter options for the algorithms. Enterprise Miner has a reasonably interesting visual interface. However, the product is hard to use especially when compared with other products such as Clementine. Enterprise Miner has powerful support for data transformation. Its visualization tools are useful for multidimensional analysis. The effort does not report any studies explicitly designed for testing the usability of the product.
NicheWorks is a tool that was originally designed for exploring large graphs [8,9]. Among other applications, the tool has been used to develop visualizations for detecting international calling frauds [10]. The detection is realized basically by using a visualization of the calling activities that allows the system user to quickly notice unusual calling patterns. The calling communities are represented using a directed graph, in which the nodes represent the subscribers whereas the links represent the calls. In particular, countries are mapped to unfilled circles, subscribers are represented by filled circles, the size and color of a filled circle reflect the total number of calls made by the subscriber. The width and color of a link represent the overall level of communication between the two ends. The tool also enables the user to drill-down on suspected patterns. It should be pointed out that, in the real sense of the word, NicheWorks is not a data mining system; it may be regarded as a visualization or exploratory tool. Therefore, the tool cannot fully accommodate the entire mining process. Nonetheless, the tool is a classic example of the role of visual data mining in visualizing raw data.

DBMiner is an on-line analytical processing (OLAP) data mining system developed by the Data Mining Research Group from the Intelligent Database Systems Research Laboratory at Simon Fraser University, British Columbia, Canada [11,12]. The system is owned by DBMiner Technology, Inc. [13]. It supports association rules, meta-patterns, classification, and clustering. DBMiner provides a browser to visualize the OLAP process. Association rules are visualized using bar charts and a three-dimensional ball graph view. With regard to visualizing decision trees, the product offers a three-dimensional graph view and an optional grid view. Clustering results are visualized using a two-dimensional graph view or a grid view. The user interface is fairly simple and standard. However, it should be pointed out that users who are not acquainted with data mining are likely to find the data mining environment somewhat intimidating. It should be acknowledged that DBMiner does interface with MS-OLAP Server and also uses MS-Excel as a browser for visualizing the OLAP process. Nonetheless, DBMiner provides no explicit support for data/results export and import. Moreover, the effort does not report any evaluation on the system.

KnowledgeSTUDIO is a product by ANGOSS Software Corporation [14]. The product supports decision trees, clustering, and neural networks. Decision trees can be constructed automatically or interactively. The product relies heavily on standard business applications (e.g. MS Office) for visualization functionalities. Due to its interactive and exploratory environment, data mining models can be produced with relative ease. Consequently, KnowledgeSTUDIO has a short learning curve. The product presently does not offer an explicit recourse for exporting data mining results (or for exchanging data mining models in general). However, it should be observed that its support for PMML could facilitate the foregoing. There is no record of any usability studies carried out on the product.

Viscovery SOMine [15] is a data mining system developed by Eudaptics. Among other data mining methods, the system supports clustering, prediction, regression, and association. The system puts complex data into some order based on its similarity. It then shows a map from which the features of the data may be
recognized and understood. On the whole, the system has a reasonably good user interface especially for self-organizing maps. The product is intended to target professional users from varied fields such as finance, marketing, industry, and scientific research. However, the effort does not report any evaluation on how successful the product is toward reaching out those types of users.

3. System architecture and implementation

We recall that from an architectural point of view, the main features of the proposed system include:

- The system is open;
- The system has a modular structure with well defined change/extension points;
- The system presents the user with a heterogeneous set of tasks in the most homogeneous and integrated way;
- The system provides the end user with the maximum flexibility during her/his data mining tasks.

At present, the system supports, but is not limited to: metaqueries [16,17], association rules [18], and clustering [19].

3.1. System architecture

The architecture of the system comprises two primary layers: the user layer, and the data mining engine layer, as seen in Fig. 1.

1) The *Parametric User Interface/User Layer* enables the user to interact with the other system components. It invokes the relevant system feature or functionality on behalf of the user. Ideally, this layer/component empowers the user to process data (and knowledge), and also to drive, guide and control the entire discovery process.

The user component is organized around a GUI container which hosts specific GUI extension cartridges and the visualization component. The extension cartridges contain the knowledge to access their respective underlying data mining components/modules in the *Data Mining Engine Layer*. In effect, the GUI container registers the specific data mining technique GUI extension, loading respective specific menu items and other commands specific to the data mining technique. For instance, the specific data mining technique GUI extension for clustering has the knowledge on how to access the clustering engine and also on how to interact with the user in the acquisition of clustering input and in the visualization of clustering output.

The GUI container provides various services to the Data Mining system. These services fall under two categories: infrastructural services and end user services.
The infrastructural services that are supported include:

- Registration of extensions which implement specific interaction contracts.
- Runtime loading on the interaction environment of the features that are relevant to the active GUI extension (e.g. commands and options).
- Advertising new GUI extensions.
- Routing user commands to the active GUI extension.

As for the end user services, the Data Mining system support includes:

- Providing the user with a uniform, consistent and flexible user interface.
- Providing services whose use span across the entire data mining interaction environment (such as start, stop, save and load services).

There are various functionalities that a data mining GUI extension supports. The GUI extension carries out the specific commands that are loaded and made available to the user (loaded on the interface) by the GUI container. The extension also implements specific input and output modalities for the underlying data mining technique or algorithm(s). On the whole, modalities specific to the data mining technique may be added or may substitute some or part of the general pre-existing ones.

(2) The Abstract Data Mining Engine Layer is completely decoupled from the User Layer. However, the structure of the Data Mining engine depicts parallelism with the structure of the GUI.
The Data Mining Engine Layer is structured using an abstract reference model based on the following concepts:

- A global dataset which contains the data to be mined and all the information necessary to apply and execute a data mining technique.
- A command manager which forms the interface between the Data Mining engine and the User Layer. On one hand, it interprets user commands originating from the user interface and on the other hand it manages any access to the internal structure of the engine. The command manager therefore serves as a two-way link between the engine and the GUI (the GUI container and the specific GUI extension). Some of the operations performed through the command manager include: defining the initial set of target data, applying some data mining algorithm, storing and verifying hypothesis.
- An abstract Data Mining verifier. This part of the engine must be specialized to implement some specific algorithm. It is worth mentioning that a verifier aims at verifying or discounting some user’s assumptions/claims about the underlying target dataset. In reality, the assumptions are hypothetical patterns and relationships regarding the data. The foregoing may be referred to as the “verification data mining goal”.
- An abstract Data Mining discoverer. The discoverer might use inputs directly given by the user or from previous data mining results. Like the abstract Data Mining verifier, the abstract Data Mining discoverer also must be specialized to implement some specific algorithm. However, it should be stated that, rather than verify some hypothetical pattern or relationship, a discoverer uses the target dataset itself to uncover or identify such patterns and relationships. The foregoing may be referred to as the “discovery data mining goal”.

The general behavior of the engine is abstract in that, it must be instantiated/specialized to specific “engines”, one for every data mining technique. It should also be pointed out that a hypothesis that is discovered and/or verified by one specific “engine” can be used by another “engine”. Consequently, the result of some data mining task can be used as input to another task. The instantiated “engines” are made available to the general engine framework dynamically. As a consequence, they are also made available to the user through the specific/respective GUI environment.

It is worth pointing out that there are some services that are available to every specific “engine”. Such services include: data management, configuration savings, intersystem communication, data access and database connection management.

As already mentioned, the architecture supports the incorporation of new and the modification of pre-existing components. Specific extension points are defined right where such component additions or modifications occur. It is envisioned that new components will be incorporated as plug-ins [2].

---

4 Intersystem communication deals with the management of the possible interactions and data transfer of different data mining techniques in a uniform way.
3.2. System implementation

In the current prototype implementation, the User Layer, which makes it possible for the user to interact with the other system components, is developed using Java and the visual interface prototype runs on Windows platforms. One of the notable features of the User Layer is the exploitation of DARE [20–22] as a visualization component/server. DARE is a visualization system that is built upon a knowledge base of rules. The system may be used to develop a visual representation of data. The importance of utilizing correct, complete, and effective visual mappings for building visualizations cannot be overestimated. It is worth mentioning that, by definition, information visualization is the process of mapping the underlying data into a visual form that will assist or trigger one to use one’s natural capabilities, mental and visual, thereby gaining insight and understanding of that data [3]. The definition itself points us to the goal of data mining. In the context of our discussion, the data to be visualized could be the target dataset or the result of a mining task. DARE is capable of analyzing whether a defined visual representation is correct, complete, and effective. In case the specified representation falls short of certain adequacy requirements, DARE is able to build and propose a more adequate visual representation.

VidaMine has a precise mapping that serves as a two-way link for exchanging information between the visual interface and data mining algorithms. In general, the mapping serves as a two-way link for exchanging information between any applications that deal with data mining aspects such as the ones hosted by the Data Mining Engine Layer, and the Data Mining Engine Layer itself. The mapping has been implemented as a definition language with corresponding syntax and semantics. In particular, the definition has been developed using XML DTDs. The reader may refer to [5] for more details on the mapping.

As it was stated in Section 3.1, the general behavior of the Data Mining engine is abstract in that, it must be instantiated/specialized to specific “engines”. In practice, the specific data mining “engines” correspond to specific data mining algorithms. Considering for instance the clustering engine in its current implementation the VidaMine system provides clustering algorithms based on the following methods:

1. Application of one of three available dissimilarity transformations (mutual neighbourhood value [23], shared neighbours count [24], average density estimate [25]), followed by searching for the largest clustering whose separation, computed as worst-case split [19], exceeds a given threshold.
2. Grouping of objects to the modes of a non-parametric kernel density estimate [26], using one of three available kernel functions (Uniform, Gaussian, or Epanechnikov’s kernel).

All methods are applicable to categorical data, using Jaccard’s distance as dissimilarity function, or to numeric data, using Euclidean distance as dissimilarity function. The multi-vantage point tree [27] is used in both cases for efficient execution of the range queries and \( k \) nearest neighbour queries needed to compute the transformed dissimilarity in method 1 and the grouping of objects to the modes
in method 2. The experimental performance of the clustering engine has been evaluated by running the clustering algorithm on synthetic datasets consisting of numeric data tuples in five dimensions, using a Pentium 4 Mobile at 1.6 GHz with 512 MB internal memory. The tuples were randomly generated following, in each dataset, a mixture of three multinormal distributions. The covariances of the three multinormal distributions were set, for all variable pairs, to 0.8, 0.85, and 0.9, respectively. The clustering algorithm was run using separation as accuracy function and the minimum function as the objectwise, clusterwise, and partitionwise quantifier. The dissimilarity transformation is the shared neighbours count. The results are shown in Fig. 2. The horizontal axis represents the number of tuples in the dataset, and the vertical axis the time in seconds to run the algorithm. The plot shows an empirical time complexity which approximates linearity.

Users of data mining systems often feel uncomfortable with the setting of numerical parameters, especially if they are not experienced or not familiar with the methods at hand. In clustering based on density estimation, the clustering engine of VidaMine applies statistical techniques to suggest a meaningful value of the smoothing parameter [26]. In clustering based on similarity or dissimilarity, the system suggests a threshold value that depends both on the dissimilarity transformation and the combination of quantifiers selected by the user. The suggested value is chosen as the middle value of the smallest interval including all threshold values yielding the most accurate clustering for a sample collection of datasets from the clustering literature.

In VidaMine clustering algorithms have been developed using Delphi. The algorithms for metaqueries and association rules have been implemented using Java.

Fig. 2. Scalability of the clustering engine with respect to dataset size.
4. The visual interface

In this section, we begin by describing how VidaMine supports the user in a data mining task as a whole. We also analyze how the system supports the user in each of the following stages: prior to the selection and application of data mining algorithms (prior to data mining model derivation), during the selection and application of data mining algorithms (during the model derivation), and after the selection and application of data mining algorithms (after data mining model derivation). Later, we demonstrate the system features by using a running example.

4.1. Supporting the user in the data mining process

The visual interface is uniform and consistent. All the visual interaction environments across different mining techniques are based on a similar design. Each visual environment comprises various visual parts/sections. Each visual part corresponds to some mining subtask such as the construction of target dataset, the selection of a data mining algorithm, the visualization of mining results, etc. The visual parts too are designed in a uniform manner across different visual environments. The uniformity aspect may be seen in the figures in this paper that show the visual interface.

Through the visual interface, it is also possible to use the results of one data mining task as input to another data mining task.

4.1.1. Prior to data mining model derivation

In this stage, there is the provision for visual preparation and manipulation of data. The data pre-processing activities should be carried out in accordance with the requirements posed by the stage and/or by the other stages. It should be mentioned that, with respect to the visual interface, this research work does not dwell much on data pre-processing aspects such as integration and cleaning. Nonetheless, as it was mentioned in Section 3, the framework on which the visual interface is based is intended to support the entire discovery process. Moreover, such aspects are part of the overall goals of the D2I project. In fact, any potential knowledge discovery research work, possibly out of the scope of the D2I project, could still be done based on (and/or easily incorporated into) the proposed framework.

VidaMine offers various visual strategies and mechanisms to support the user in carrying out activities prior to the selection and application of data mining algorithms, such as the ones described in the sequel.

4.1.1.1. Visual construction of the dataset. The user directly interacts with data (the actual data and other relevant information such as metadata or previous mining results) in specifying a set of task-relevant data for any of the supported data mining

---

5 It is worth mentioning that the main challenge at this lower level of uniformity stems from the sometimes rigid and specific data mining algorithm parameters. The challenge would impact primarily the data mining modeling part of the visual interaction environment.
strategies. In the proposed system, the target dataset may comprise one or more relations. The task of constructing/defining the target dataset relies on two main interaction components namely the Specification Space and the Target Space. These may be seen in all the figures illustrating the input environments of the visual interface (e.g. in the top part of Fig. 4).

The Specification Space supports the user in specifying a set of task relevant data. The component acts as the workshop from which the user “constructs” the desired relation. The Target Space acts as a container for the relations that the user specifies. The two interaction components are complementary. Therefore, the user operates by appropriately moving between the two components.

4.1.1.2. Visualization of raw data. The support for the visualization of raw data is especially valuable to the casual/novice user (i.e. a user who is not acquainted with data mining concepts) and even to a starter who may not be having adequate or any knowledge about the data at hand. Such users could simply visualize the current data, probably attempting various visualization styles and mappings. This would make the user to get acquainted with the data at hand and boost her/his confidence in using the data and the system. This is a step forward in ensuring that the system supports casual (and uninformed) users. In fact, even expert users would be in a far much better position to apply to the dataset the data mining techniques supported by the system, if they first got the opportunity to get acquainted with that dataset.

Through the DARE [20–22] visualization component/server, VidaMine produces a visualization of raw data. Fig. 3 shows an example of a visualization that allows the user to explore raw data. A tuple in the dataset is represented by a point. The attributes of a tuple are mapped as follows: CustID is mapped to the $x$-axis, ServID is mapped to the $y$-axis, CentID is mapped to the $z$-axis, CustGender is mapped to color and CustAge is mapped to size.

4.1.2. Data mining model derivation

Model derivation involves activities such as the visual specification of the model and its parameters, and visual support for the storage of results.

Toward enabling the user to derive a data mining model, VidaMine employs various visual strategies. The user can visually construct the mining query. For instance, in the:

4.1.2.1. Metaquery environment. In this environment, the user can specify patterns/relationships between or among data tables in an intuitive visual manner. The interface provides “hooks” and “chains” through which the user can visually specify the relationships. By simply linking two attributes, the user indicates to the system that s/he is interested in metarules that have the two attributes related. Therefore, VidaMine enables the user to visually construct metaqueries of interest. Fig. 6 shows the part of the metaquery environment that supports the foregoing specification.

4.1.2.2. Association rule environment. In the environment, there is the provision of “market baskets”. As seen in Fig. 9, the association rule environment offers two
baskets, the IF basket and the THEN basket. The IF basket represents items in the antecedent part of an association rule and the THEN basket represents items in the consequent part of the rule. The user may “drag and drop” items from the target dataset into the relevant baskets. When through with the visual construction of a particular association rule, the user may empty both baskets into a pool of association rules that are of interest to her/him, and may go ahead and construct another rule.

4.1.2.3. Clustering environment. In this interaction environment, there are various visual widgets for specifying or selecting parameters characterizing a clustering task including:

- A fixed number of clusters or a measure of homogeneity, separation or density.
- Attributes that will be directly involved in cluster analysis.
- Supplementary attributes e.g. for labeling cases in the output.

Specifying each such parameter may also involve more specific options/settings. The environment can be seen in Fig. 4.

![Fig. 3. Visualization of raw data.](image)
4.1.3. After data mining model derivation

Data mining algorithms are often able to handle large amounts of data. However, the size of the display is fixed and limited. Be that as it may, the results of data mining algorithms often are in a form that is difficult to be understood by humans who are accustomed to perceiving information by their visual senses. However, through the appropriate use of effective visualizations, all relevant or at least much of the relevant data can be represented in an understandable manner.

Besides exploiting DARE [20–22] to visualize raw data, the visualization component can also be used to develop a visualization for some mining result. With regard to clustering, some of the mining result aspects whose meaning should be made apparent include: density, homogeneity and separation measures. Regarding metarules and association rules, aspects of interest include: measures of interestingness (such as confidence and support) and the items participating in a particular rule. For example, Fig. 8 shows a visualization of metaquery results. In the visualization, a rule is mapped to a point, the confidence value of a rule is represented by size, the support value of a rule is represented by the y-value and the number of items in a rule is represented by the x-value.

It is interesting to observe that the visualization component can be used to carry out a visual exploration of the results. Moreover, the mining results or a subset of the same could be “exported” to serve as the set of target data for another mining task. The foregoing functionalities are especially resourceful when it comes to post-processing activities.

4.2. Demonstration

Toward describing the system features, we consider as running example, a communication company that provides various services such as Web-access services, telephone services, etc. The company has a main office and a number of service centers. The main office principally deals with strategic and administrative issues. In fact, the company offers its services through the service centers. The company plans to introduce some special offers. The company’s marketing strategist is expected to recommend the type of service to be featured and the customers to be targeted. Assume that the marketing head decides to mine some information using VidaMine. In this case, we may view him as the user of the system.

The marketing strategist might want to identify regions that had relatively good general service sales in the past. He might want to use that information further to propose some specific service and customers that might be worth consideration in the offer. The recommendation could also include another service that normally does best with the proposed service.

4.2.1. Identifying regions with good sales: using the clustering environment

Understanding how different regions have been doing can be resourceful in making marketing decisions. The task can be accomplished through the clustering environment.
As a user, one starts by specifying a target dataset by using the Specification Space and the Target Space. In this task, the marketing lead is mainly interested in customers and services (i.e. based on relations Customer and Service or on relation CustServ). The company already has geographical information pertaining to customer addresses. For instance, their loci with respect to the main office. The user may construct a relation in which the attributes of interest are CustID, CustX, CustY, CustAmt and ServID.

VidaMine provides an interaction environment with various input widgets through which the user can specify parameters characterizing a clustering task. The user can specify:

1. A measure of homogeneity, separation or density. Alternatively, the user may specify a fixed number of clusters. There are radio-buttons to enable the user to make the selection between the measure and the fixed number of clusters. Moreover, there is a radio-button for each of the three measures. The user can express interest in a measure by selecting the corresponding radio-button. Each of the measures is presented on an ordinal scale that runs from 0 to 100. There is a slider control for each scale. The user can set a measure by dragging the slider to some value/position of interest on the scale. It should be pointed out that the Similarity slider on the interface corresponds to the homogeneity measure whereas the Dissimilarity slider corresponds to the separation measure. With regard to supporting the specification of a fixed number of clusters, the interface provides spin-boxes (and alternatively an edit-box). With any of the foregoing two main clustering options, the user may also specifically define objectwise, clusterwise and partitionwise quantifiers in terms of minimum, maximum, sum and average operators. The quantifiers have to do with homogeneity or separation. As for density, the respective combo-box enables the user to specify the kernel function whereas setting the respective slider corresponds to smoothing the clusters. As seen in Fig. 4, the user has specified a separation measure and the corresponding quantifiers.

2. Attributes that will actively/directly participate in cluster analysis. The interface supports a “checking” mechanism for specifying such attributes. A relation in the Target Space possesses features such as a “handle” and a “check-box” for each attribute. “Checking” a “check-box” means that the corresponding attribute has been chosen to directly participate in cluster analysis. On the other hand, the user may undo (“uncheck” attributes) the operation. These operations are realized through standard “point-and-click” functionalities. In Fig. 4, the attributes CustID, CustX and CustY have been selected (“checked”).

3. Attributes for supplementary purposes. In particular, the user may specify an attribute to be used in labeling cases in the output.

When through with the parameter specification, the marketing strategist may instruct the system to partition the target dataset by clicking the “torch” icon. The system performs the clustering and displays the results.
With respect to displaying the output of a clustering task, VidaMine uses the visualization component. The system supports two main visualization mechanisms: **Clusters + Details ("Overview + Detail")** and **Dedicated View**.

### 4.2.1.1. Clusters + Details ("Overview + Detail")

This visualization displays clusters on a scatter plot, and also presents details that correspond to a selected cluster or cluster object. The former display corresponds to an "Overview" window whereas the latter corresponds to a "Detail" window.

Cases are mapped to points on the scatter plot, with each point taking some $x, y$ (and if appropriate $z$) values. A currently selected cluster, cluster object, or outlier is drawn with an outline around it. The points may be encoded to reflect other aspects (e.g. by using color and size). The top-right part of Fig. 5 shows the visualization in which the values on the $x$-axis correspond to $\text{CustX}$, those on the $y$-axis correspond to $\text{CustY}$ and the $z$-axis values correspond to $\text{CustID}$. The "Detail" window is an exposition of a selected cluster or point.

### 4.2.1.2. Dedicated View

Homogeneity, separation and density measures are useful in many ways (e.g. for interpretability and evaluation purposes). In the clustering environment, the **Dedicated View** displays the measures of homogeneity, separation and density for each and every cluster or outlier. The separation measure is mapped to the $y$-axis. A circle encodes a cluster or an outlier. The circles are arranged along the $x$-axis. The density of the cluster or outlier is bound to the diameter of the circle. The grayscale level of a circle represents the homogeneity measure of the represented cluster or outlier.

From the **Clusters + Details** and **Dedicated View** visualizations of Fig. 5, regions that are close to the main office depict a lot of sales. With regard to the anticipated offer, a possible marketing strategy would involve putting a lot of emphasis on people and service centers that are close to the main office.

The marketing strategist might want to gain more knowledge from those interesting regions. For instance, he might want to identify some specific service and customers within those particular regions. The task would entail establishing data relationships. The analysis can be done through the metaquery environment.

However, it is important to observe that the task is based on some particular subset of data which is not equivalent to the currently defined set of target data. In other words, the user intends to use some output from one task (clustering) as an input to another task (metaquerying).

The interface enables the user to select points or clusters of interest through the use of the **Standard Tools** toolbox. The marketing strategist may then turn to the **Export Facility**. The facility would enable him to specify whether he would want to just save the specified output or to save and switch to another task with that output as the input to the new task. In the latter case, the system switches to the new environment with the output appearing in the Specification Space as a resource relation.
4.2.2. Establishing data relationships: using the metaqueury environment

The marketing strategist would need to analyze the relationships that exist among services, customers, and centers. The analysis would help him to determine the
Fig. 5. The clustering environment: output.
service to feature and potential customers. The metaquery environment can be helpful in carrying out the analysis. Such relationships can be mined by exploiting the relations $\text{CustCent}$ (customers vs service centers), $\text{ServCent}$ (services vs service centers), and $\text{ClustOut1}$. It should be observed that the latter relation, $\text{ClustOut1}$ was “imported” from the clustering task and it relates to customers vs services. The intended effect is to have the metaquery analysis restricted to only the tuples contained in the data that was “imported” from the previous task (tuples in the relation $\text{ClustOut1}$).

Consider that the user is specifically interested in the following attributes: $\text{CustCent.CustID}$, $\text{CustCent.CentID}$, $\text{ClustOut1.CustID}$, $\text{ClustOut1.ServID}$, $\text{ServCent.ServID}$ and $\text{ServCent.CentID}$. Therefore the marketing strategist needs to specify the three relations with the foregoing attribute constraints toward constructing the target dataset. In Fig. 6, the marketing strategist has already constructed each of the three relations with the respective attributes and dropped each into the Target Space.

In the environment, the user may define links manually (Manual Links) or have the system automatically do that (Automatic Links). (A link defines a connection/relationship between attributes that is aimed at generating a consequent pattern.) Assume that the marketing strategist chooses the latter option. The system links the attributes as follows:

- $\text{CustCent.CustID}$ with $\text{CustServ.CustID}$
- $\text{CustCent.CentID}$ with $\text{ServCent.CentID}$
- $\text{ServCent.ServID}$ with $\text{ClustOut1.ServID}$

Letting $X$ be a representation for $\text{CustID}$, $Y$ for $\text{CentID}$, and $Z$ for $\text{ServID}$, and allowing reordering of attributes, the system generates the following transitive “combinations” (which are actually metaqueries):

1. $\text{CustCent}(X, Y), \text{ClustOut1}(X, Z), \text{ServCent}(Z, Y)$
2. $\text{ClustOut1}(X, Z), \text{CustCent}(X, Y), \text{ServCent}(Y, Z)$
3. $\text{ServCent}(Z, Y), \text{ClustOut1}(Z, X), \text{CustCent}(X, Y)$

The system puts the generated metapatterns in a “pool” (of metaqueries), as seen in Fig. 6. The foregoing generation and putting of metapatterns in the “pool” are realized by clicking the Add Pattern command button. The strategist may also specify confidence and support values by using sliders or edit-boxes. He may then instruct the system to search for specific rules from the target dataset that correspond to the metapatterns in the “pool” and that satisfy the specified parameters, by clicking the “torch" icon.

Through the visualization component, VidaMine provides various visualizations for the search results. For any rule, the aspects that are of principal interest include: measures of interestingness, relationship between the head and body, and details about the items participating in the rule. The system provides two main visualizations for the search results Rules + Tuples (“Overview + Detail”) and Dedicated View.
Fig. 6. The metaquery environment: input.
4.2.2.1. *Rules+Tuples* ("Overview + Detail"). This visualization displays all the rules from the search operation, and also presents tuples that correspond to some selected rule(s).

The rules display (scatter plot) may be envisaged as the "Overview" window and the tuples display as the "Detail" window. In visualization, it is crucial that conceptually important aspects or dimensions be made perceptually important. With regard to rules, their (relative) confidence and support values are conceptually important. In the scatter plot, the two variables have been mapped to grayscale and y-axis respectively. On the whole, the scatter plot mapping is as follows: a rule is represented by a point, the confidence value of a rule is represented by grayscale, the support value of a rule is represented by the y-value, and the number of items participating in a rule is represented by the x-value. Consider that the metaquery search based on the ongoing example returns the following results:

1. \( \text{CustCent}(\text{CustID}, \text{CentID}) \rightarrow \text{ClustOut1}(\text{CustID}, \text{ServID}), \text{ServCent}(\text{ServID}, \text{CentID}) \)
   - Support = 33.33% and Confidence = 100%
2. \( \text{ClustOut1}(\text{CustID}, \text{ServID}) \rightarrow \text{CustCent}(\text{CustID}, \text{CentID}), \text{ServCent}(\text{CentID}, \text{ServID}) \)
   - Support = 44.44% and Confidence = 75%
3. \( \text{ServCent}(\text{ServID}, \text{CentID}) \rightarrow \text{ClustOut1}(\text{ServID}, \text{CustID}), \text{CustCent}(\text{CustID}, \text{CentID}) \)
   - Support = 55.56% and Confidence = 60%

In Fig. 7, there is a *Rules + Tuples* visualization of the foregoing results. In the visualization, the marketing strategist has selected the rule with the highest confidence for exposition. The tuples window depicts some interesting trends. The service represented by black circles had the highest demand. The marketing lead may interact with the display, for instance by using the exposition tools or by pointing the circle(s), thereby getting to know that the interesting service is *WWW-Access*. The display also depicts that, virtually all the customers who requested the *WWW-Access* service are young.

Consequently, the marketing strategist may wish to consider *WWW-Access* service for the anticipated offer. Moreover, he has fairly substantial information regarding the customers to target: those living close to the main office and who are of young age.

4.2.2.2. *Dedicated View*. Like the foregoing visualization, the *Dedicated View* enables the user to visualize all the rules from the search operation. However, in this case, rules are visualized in a more elaborate manner. The *Dedicated View* displays: the confidence and support values of each rule, the relationship between the head and the body of each rule, and the individual items/components that make up each rule. The visualization uses a simple 2D floor with a perspective view. The floor has rows and columns. A rule is represented by a column on the floor. The rule is made up of the components which have entries in the column. The rows represent the
Fig. 7. The metaquery environment: output.
items (such as attributes). Associated with each column/rule is a “bucket”. The gray value of the contents of the “bucket” represents the confidence value of the rule. The level of the contents of the “bucket” is bound to the support value of the rule. The handle of the “bucket” can be used to select the corresponding rule. Rule items that form the antecedent are each represented using a “key” icon, whereas those that form the consequent are each represented using a “padlock” icon. The visualization can be seen in the bottom part of Fig. 7.

One of the problems with a perspective view is the determination of size of far objects. Nonetheless, in our visualization, the icons/“objects” on the floor represent either the antecedent or the consequent of a rule. As such, their size is not a major issue in interpreting the visualization as long as they are visually distinguishable as representations of the head or of the body.

By specifically exploiting DARE [20–22] as the visualization component/server, various visualizations can be realized. For instance, Fig. 8 corresponds to Fig. 7’s visualization of rules and in which: a rule is mapped to a point, the confidence value of a rule is represented by size, the support value of a rule is represented by the y-value and the number of items in a rule is represented by the x-value.

![Fig. 8. Visualization of metarules.](image-url)
4.2.3. Market-basket analysis: using the association rule environment

The marketing strategist might also intend to find out another service that is frequently requested every time WWW-Access service is requested. Such knowledge would be instrumental in making some marketing decisions. For instance, in designing advertisements that capture the two products. The analysis can be realized by switching to the association rule environment. It is worth mentioning that, if the user were interested in switching to the new environment with the previous output as the new input, he could use the Export Facility that was described at the end of Section 4.2.1. One of the distinct features in the association rule environment is the provision of “market baskets”. Although the relation Transaction is required as part of the target dataset, in the association rule environment it is enough for the marketing strategist to just interact with the relation Service without having to understand the transaction details. Consider that the relation Service has attributes Service.ServID and Service.ServName. In Fig. 9, the marketing strategist has specified the two relations as the target dataset. Toward specifying the structure of an association rule of interest, the user drags tuples from the target dataset and drops them into either the first (“IF”) basket or the second (“THEN”) basket. Recalling that the marketing strategist is interested in WWW-Access, he would therefore drag and drop the tuple Service = “WWW-Access” into the first “basket”. He may leave the second “basket” empty as a generic service entry that the system will later instantiate with various relevant service entries. The user then empties the baskets into the “pool” by clicking the icon marked with tilted baskets. As seen in Fig. 9, the effect of having left the second basket empty is apparent in the “pool” in that the “THEN” part of the rule is generic (some variable “X”). The user may specify confidence and support measures. He may then instruct the system to search (and display) specific rules from the target dataset that correspond to the association rule structure(s) in the “pool” and that satisfy the specified parameters.

The returned association rules are visualized using the same mechanisms that are used for visualizing metarules. As seen in Fig. 9, the marketing strategist has performed some basic selection of the association rule with the highest confidence in the scatter plot. Tuples corresponding to the rule are displayed in the Tuples display. It is worth mentioning that by observing the particular association rule, the marketing strategist may be able to determine the best service to associate with WWW-Access (such as Printing).

5. Usability tests

ISO 9241-11, Ergonomic requirements for office work with visual display terminals: Guidance on usability (1998), defines usability as “the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use”. ISO 9241 also identifies the following as the most useful indicators in measuring the level of usability of a product:

- Effectiveness in use, which encompasses accuracy and completeness through which users achieve certain results.
Fig. 9. Association rule environment: input.
Efficiency in use, which has to do with the resources utilized in relation to accuracy and completeness.

Satisfaction in use, which includes freedom from inconveniences and positive attitude toward the use of a product.

The new ISO/IEC 9126-1 definition of “goal quality” bears resemblance to the foregoing definition. Consequently, usability has an impact on the quality of a product. The evaluation of goal quality includes the use of effectiveness, efficacy and satisfaction indicators, and security [28].

Various usability tests have already been designed and carried out on the system during the development life cycle. The input and recommendations of the users-to-be were instrumental in guiding the user interface design process. Undertaking such measures can shield the system from exhibiting drawbacks after it becomes operational.

VidaMine targets two categories of users: expert users and casual users. In the former category, we have users who are acquainted with data mining such as data miners, statisticians, and data analysts. In the latter category, we have users who have little or no knowledge in data mining such as managers. The user audience factor was taken into consideration by involving user representatives from both categories in the experiments.

As a way of getting started, we carried out a heuristic evaluation [29] on an initial proposal of the Data Mining user interface. Heuristic evaluation refers to a more informal evaluation where the interface is assessed in terms of more generic features. This informal evaluation presents reasonably concise and generic principles that apply to virtually any kind of user interface. In the sequel is an analysis of how some of the principles have been applied in the design of the Data Mining system.

- The interface dialogue should be simple and natural. Moreover, the interface design should be based on the user’s language/terms. In general, there should be an effective mapping between the interface and the user’s conceptual model. In VidaMine, the user interface primarily uses basic data mining terms. Furthermore, the provision of “hooks” and “chains” for linking attributes, “baskets” for designing association rules, “drag and drop” mechanisms, “buckets” for measures of interestingness, and “keys” and “padlocks” for antecedents and consequents are part of the effort aimed at getting effective mappings between the interface and the user’s conceptual model.

- The interface should shift the user’s mental workload from the cognitive processes to the perceptual processes. Our Data Mining interface supplies various mechanisms to support the shift. For practically all inputs, the user does not have to supply the units of measurement. Moreover, the system offers interaction controls (e.g. sliders) for helping the user get familiar with the range of valid values and also for helping him/her input within the range. Furthermore, visually presenting query parameters (e.g. data relations) minimizes the possibility of making mistakes while formulating a query.

- There should be consistent usage and placement of interface design elements. Consistency builds confidence in using the system and also facilitates exploratory learning of the system. In our interface, the same information is presented in the
same location on all the screens. In fact, the Data Mining interface is uniform across the various environments for metaqueries, association rules, and clustering.  

- The system should provide continuous and valuable feedback. One of the mechanisms our Data Mining system uses to provide feedback is realized when the user puts some item into the “baskets” or empties the “baskets”. The “baskets” respond to reflect the insertion or the removal. Moreover, the visualization dynamically updates itself as the user changes (or interacts with) data mining parameters.  

- There is a need to provide shortcuts especially for frequently used operations. In the Data Mining interface, there are various shortcut mechanisms. For instance, double clicking and single key press commands. The anticipated incorporation of a visual query language is expected to also help with respect to shortcuts.  

- There are many situations that could potentially lead to errors. Adopting an interface design that prevents error situations from occurring would be of great benefit. In fact, the need for error prevention mechanisms arises before (but does not eliminate) the need to provide valuable error messages. Our Data Mining interface offers mechanisms to prevent invalid inputs (e.g. specification by selection, specification through sliders). It also provides some status indicators (e.g. when an item is put in a “basket”, the status of the “basket” changes to indicate containment).  

Moreover, we initially designed a mock-up interface. We presented it before the whole team of data mining experts who are participating in the D2I project and carried out a simulation of their typed tasks. We got encouraging results from the tests and even suggestions on how to improve the interface. For instance, the data mining experts suggested that the interface should allow the user to specify more generic relations when constructing the set of target data. They also suggested that the interface should offer the possibility to put constraints on such a generic relation (e.g. by specifying the number of its attributes). Consequently, the foregoing suggestions have been included by introducing the Specification Space and the Target Space in the current design of the interface. The data mining experts also suggested that the system should provide an optional interaction environment specifically designed for the expert user and still leave the user with the freedom to switch between the two.  

Recently, we performed some more user tests on the current version of the prototype with five selected users from the universities of Bologna, Ferrara, and Modena and Reggio Emilia.  

The selection of the subjects was done in such a manner that both expert users and casual users were represented. One of the tested users is a multi-media post-doctorate student with some basic knowledge in data mining. Another user is an image analysis graduate with very little knowledge in data mining. Two of the users are data mining researchers. The last user is a post-doctorate student in pattern recognition with some little knowledge in data mining.  

The experiment involved: the prototype as an application, a case study, user tasks corresponding to the case study, data schema corresponding to the data used in the
case study, and a questionnaire. Each user was expected to run and interact with the interface of the prototype with reference to the accompanying documents (case study, user tasks, and data schema). After the experiment, the user would fill in the questionnaire.

The first part of the questionnaire contained closed questions pertaining to the simplicity or complexity of carrying out user tasks. At least three of the users found each of the supported tasks fairly easy to carry out. In particular, all the users found the clustering output very easy to understand and interact with. Four of the users found specifying target dataset and clustering input very easy to perform. Two of the users found it very easy to construct metaqueries and association rules. Exactly three users found it fairly easy to carry out and interact with the visualization(s) of metarules and association rules. Exactly three users also found it fairly easy to perform a mining task in general. As for using the output of a particular task as the input to another task, exactly two users found it fairly easy to perform.

The second part of the questionnaire also had closed questions. The part was aimed at assessing interface design aspects. At least three of the users found each of the tested design aspect reasonably well-adhered to. Three of the users observed that consistency was very well-applied in the interface design. All the five users found the interface perceptiveness/intuitiveness fairly well-adhered to. Four of the users felt that the user’s language was fairly well-applied. Four of the users noted that it was fairly easy to remember (or acquire) the aspects relevant to a particular mining task. Moreover, the same percentage observed that the system responded to user operations in a reasonably valuable way. Three of the users found the interface elements fairly well organized.

The last part of the questionnaire contained open questions pertaining to strengths, weaknesses, capability of the system/interface. It also had room for extra/other comments.

It is interesting to realize that for the most-liked interface features, some of the subjects highlighted the same features. Two users mentioned consistency as one of the main features they liked most about the interface. Two users mentioned good layout/organization as one of the main features they liked most about the interface. Two users mentioned the support for visual exploration as one of the main features they liked most about the interface. Therefore, the foregoing indicates that the interface is strong in consistency, layout/organization and visual exploration.

On the other hand, some of the users stated similar interface aspects that they considered to be negative. Two users found the size of some of the visual elements small/big or allocated little/a lot of space. This could be linked to some users’ observation that there are many visual elements on the interface. Based on one subject’s statement of a negative aspect and another subject’s comments, it is apparent that the role of the visual control Number of Attributes in the Specification Space is not clear. Therefore, the foregoing issues should be carefully considered in the development of the future versions of the prototype.

As for the functionalities supported by the system, three of the users were satisfied. However, two of the users did recommend that the system should provide an explicit and visual way of expressing joins in the Specification Space.
At present, we are modifying the current version of the prototype based on the results obtained from the foregoing user studies. After that, we will run formal usability experiments on the revised version. Consequent results would be instrumental in determining further relevant interface improvements and modifications.

6. Future work and conclusions

Modifications and improvements on the current version of the prototype on the basis of the results obtained from the questionnaire experiments, discussed in Section 5, are going on. Once realized, the revised prototype version will be subjected to formal usability studies.

With regard to clustering, there are plans to support the output of descriptive statistics principally using pie-charts and histograms. Moreover, a set of the system features are under an engineering process in order to become part of the commercial suite D2I, distributed by Inspiring Software (http://www.cisat-group.com). The D2I (Data to Information) environment is thought for handling the data needed by a generic process. As an example of real application, in Fig. 10, about 20,000 postal parcels are visualized.
parcels are shown, in order to analyze the parcel dimension distribution with respect to the parcel weights. In particular, length, width, and height correspond to $x$, $y$, and $z$ axes, respectively, while the weight is associated with the color coordinate (blue dots represent light parcels, red dots represent heavy parcels) allowing the user to perform a visual data mining activity. Presently, Inspiring Software is including in the next commercial releases part of the clustering issues described in this paper.

In this paper, the need to develop knowledge discovery systems based on an open framework has been highlighted. The importance of providing an effective visual interface to support the user across the entire process of mining knowledge has been pointed out. The paramount role of usability studies and formalization has been discussed. The paper has proposed and described VidaMine, a visual data mining system that has been designed with the foregoing issues taken into consideration.

References


