Employing Inductive Databases in Concrete Applications

Rosa Meo[1], Pier Luca Lanzi[2], Maristella Matera[2],
Danilo Careggio[1], and Roberto Esposito[1]

[1] Università di Torino, Dipartimento di Informatica,
corso Svizzera 185, I-10149, Torino, Italy
{meo, careggio, esposito}@di.unito.it
[2] Politecnico di Milano, Dipartimento di Elettronica e Informazione,
Piazza Leonardo da Vinci, 32, I-20133, Milano, Italy
{lanzi, matera}@elet.polimi.it

Abstract. In this paper we present the application of the inductive database approach to two practical analytical case studies: Web usage mining in Web logs and financial data. As far as concerns the Web domain, we have considered the enriched XML Web logs, that we call *conceptual logs*, produced by specific Web applications. These ones have been built by using a conceptual model, namely WebML, and its accompanying CASE tool, WebRatio. The Web conceptual logs integrate the usual information about user requests with meta-data concerning the Web site structure. As far as concerns the analysis of financial data, we have considered the trade stock exchange index *Dow Jones* and studied its component stocks from 1997 to 2002 using the so-called *technical analysis*. Technical analysis consists in the identification of the relevant (graphical) patterns that occur in the plot of evolution of a stock quote as time proceeds, often adopting different time granularities. On the plots the correlations between distinctive variables of the stocks quote are pointed out, such as the quote trend, the percentage variation and the volume of the stocks exchanged. In particular we adopted *candle-sticks*, a figurative pattern representing in a condensed diagram the evolution of the stock quotes in a daily stock exchange. In technical analysis, candle-sticks have been frequently used by practitioners to predict the trend of the stocks quotes in the market.

We then apply a data mining language, namely `MINE RULE`, to these data in order to identify different types of patterns. As far as Web data is concerned, recurrent navigation paths, page contents most frequently visited, and anomalies such as intrusion attempts or a harmful usage of the resources are among the most important patterns. As far as concerns the financial domain, we searched for the sets of stocks which frequently exhibited a positive daily exchange in the same days, so as to constitute a collection of quotes for the constitution of the customers' portfolio, or the candle-sticks frequently associated to certain stocks, or finally the most similar stocks, in the sense that they mostly presented in the same dates the same typology of candle-stick, that is the same behaviour in time.

J.-F. Boulicaut et al. (Eds.): Constraint-Based Mining, LNAI 3848, pp. 295–327, 2005.

The purpose of this paper is to show that the exploitation of the nuggets of information embedded in the data and of the specialised mining constructs provided by the query languages, enables the rapid customization of the mining procedures following to the users' need. Given our experience, we also claim that the use of queries in advanced languages, as opposed to ad-hoc heuristics, eases the specification and the discovery of a large spectrum of patterns.

1 Introduction

In this paper we present two case studies, the mining of the Web log of a University Department and the analysis of financial data, Dow Jones index of the market stock exchange in a long period of time. We conducted the analysis by means of the exploitation of a query language for association rule mining – `MINE RULE` – and used it in a fashion typical to inductive databases. This choice has two purposes: (i) analyzing the concrete domain data and extracting interesting, usable and actionable patterns; (ii) evaluating the usability and expressive power of the mining query language by itself and inside the context of an inductive database approach.

For the analysis of Web domain, we have adopted *WebML conceptual logs* [11], that are Web logs enriched with novel information with respect to standard Web logs. Indeed, they have been obtained by integration of (ECFL) Web server logs with information on the Web design application and information on the Web pages content.

For the analysis of the financial domain, we adopted technical analysis, a typology of analysis that relies upon the study of the quotes plots in their temporal evolution. In particular, we adopted a typology of pattern named *candle-stick*. We searched in the data many configurations of these patterns, and some of them have been considered for a long time by the practitioners of the field a predictive tool for the determinant changes in the market stock quotes.

Inductive databases were proposed in [14] to leverage decision support systems by means of the application of specific mining languages to the analysis of data stored in databases. Thus, the user/analyst can interact with the system during the knowledge discovery process in a similar way as with a DBMS, that is, by specifying interesting data mining patterns with queries that in turn retrieve and store in the database the discovered information.

The adoption of this case study had also purpose to verify and experiment the suitability of some knowledge discovery (KDD) scenarios for inductive databases. KDD scenarios have been produced as a set of characteristic queries solving some frequently asked questions (mining problems) by users/analysts in order to recover from frequently occurring problems. We devolped some of these KDD scenarios inside the *cInQ* project (http://www.cinq-project.org), an european funded project and a consortium of universities and industries launched in order to evaluate the feasibility of inductive databases and of mining languages to extract interesting and actionable patterns from real data-sets.

The case studies are nowadays domains of extreme importance.

For the first case study, the analysis of Web logs, we identified three main typologies of user needs for which the association rules could constitute an aid: (i) the identification of frequent crawling paths by the users (Web structure and Web usage analysis), (ii) the identification of user communities and (iii) the identification of critical situations (anomalies, security attacks, high traffic) in which the information system could be placed.

The first topic allows the customization and construction of adaptive Web sites and recommender systems. The second one is important for customer relationship management and business applications (e.g, e-commerce). The third one is essential to the management of computer security for a reliable, efficient and available information system and its Web front-end and also to the credibility of Internet iteself.

To solve the first need the association rules that are searched for are those ones that identify sets of pages most frequently accessed by a consistent number of users. Furthermore, these sets of pages could have related information content. In these cases the discovered crawling paths could identify the most valuable information content and therefore could suggest a potential restructuring of the design of the Web site (e.g., by making easier the search for those pages by the creation of ad-hoc indexes or by fast access paths). This aspect is strictly correlated to the construction of a dynamical and adaptive Web site that is able to learn by previously submitted requests, changes accordingly to the user in order to fit best to its probable, future requests. Furthermore, the acquired knowledge allows the enhancement of the Web site with the potentialities of a recommender system, that suggests the user the preferable paths and interesting contents he/she could search for, on the basis of the frequently observed user visits and of the user profiles.

The second topic is very important for customer relationship management and many business applications, e.g., e-commerce applications and targeted marketing.

The third topic is also very important and the application of data mining in this field could give a consistent aid in information system administration as well as in computer security management. Indeed, nowadays we continuously observe the verification of dangerous situations in which our information systems are placed under attacks or are blocked by a highly congested traffic. Therefore, the searched association rules will try to give a descriptive profile of situations that frequently end in generation of errors by the Web server (situations that mostly correspond to hackers' attacks) such as repeated sequences of operations or services or the usage of a certain browsers or operating systems. Another kind of useful association rules tries to provide a profile of critical users, either because they request frequently a large volume of data or because they are often associated to certain typologies of errors returned by the system.

We had in addition the motivations to augment the quality of a Web application site, which is nowadays very important for a company (especially for e-commerce applications) since it reflects the immage of the company and might constitute its success or failure.

The second case study, the stock trading, had also very concrete motivations. Of course, the analysis of financial data and a better understanding of the stocks market behavior and evolution are of extreme importance in stock trading: it might allow investors to regain faith in the market, reducing the investors' risks, their losses and increasing the gains. An efficient data mining activity on this type of data is very important, since nowadays trading mainly occurs on-line and real-time, and is a very demanding process since it requires the real time operativity from every internet access point on huge volumes of stream data. Furthermore, the capability to make in advance the right prediction has enormous importance since the negotiation activity of stocks is a very rapid process, and involves huge flows of money: in this context, the right operation at the right time can make the investors gain or save huge volumes of money.

In this context, the main user needs are coarsely the following.

(i) Detect the best performing stocks in order to compose the portfolio. Possibly this consists also in the identification of the most similar stocks, once again, in order to acquire better knowledge on the stocks quotes behavior.

(ii) Detect the patterns configurations, in the stock quotes, that most frequently were associated to particular situations in the market, such as a high instability, and allowed to predict a consistent variation in stocks quotes.

(iii) Verification of some principles in technical analysis, principles that usually have acquired the consent of practitioners, but not always have gained the approval of statisticians and experts in economics. One of some principles, originally proposed by Dow Jones, but also confirmed by the most rigorous economists is the fact that a consistent increase in the volumes exchanged of a stock corresponds also to a notable variation in the prices of the same stock.

The paper is organized as follows. Section 2 provides an overview of the first application presented: the analysis of Web logs. Section 3 describes the `MINE RULE` operator briefly and the Web Log case study. Section 4 describes the second application: the analysis of financial data by the adoption of concepts of technical analysis (candle-sticks). In these latter Sections we provided also the KDD scenarios made of queries that allowed us to obtain useful results, and for each of them we provided a detailed description. Sections 5 and 6 respectively provide an evaluation of obtained results and some guidelines of using inductive databases in the analyzed application domains. Finally Section 7 draws the conclusions.

2 Application 1: Web Log Analysis

Current Web applications are very complex and highly sophisticated software products, whose quality, as perceived by users, can heavily determine their success or failure. A number of methods have been proposed for evaluating the quality of Web applications. In particular, Web usage mining methods are employed to analyze how users exploit the information provided by the Web site. For instance, showing those navigation patterns which correspond to high Web usage, or those which correspond to early leaving [17,27].

Web usage mining is mainly performed on the server side, and therefore is based on information found in log files, containing data about single user page access requests, represented in one of the available standard formats [25,4,25]. Accordingly, Web Usage Mining approaches rely heavily on the preprocessing of log data as a way to obtain high level information regarding user navigation patterns and ground such information into the actual data underlying the Web application [26,9,5]. Preprocessing generally includes four main steps:

- *Data cleaning*, for removing information that is useless for mining purposes, e.g.: requests for graphical contents, such as banners or logos; navigations performed by robots and webspiders.
- *Identification of user sessions*, for extracting full users navigation paths from the poor information available in Web logs. This step can be very demanding [26], especially due to the adoption of proxy servers by applications, which do not allow the unique identification of users from Web logs.
- *Content and structure information retrieval*, for mapping users page requests into the actual information of visited pages.
- *Data formatting*, for preparing data obtained through the previous three steps for being processed by the actual mining algorithms.

Notwithstanding the preprocessing effort, in most of the cases the information extracted is usually insufficient and with much loss of the available information at the Web design level, such as the complete knowledge about the structure and content organization of a Web application [23,6].

The approach we present in this paper is different with respect to other methods so far proposed, in that Web Usage Mining is directly integrated into the Web application development process. We use a conceptual model for Web application design (*WebML*) and its supporting case tool (*WebRatio*) for the development of Web applications that are able to produce rich Web logs (*conceptual logs*). Conceptual logs provide the mining phase with some of the necessary information with no loss and no additional cost: the content and hypertext structure of the application (originally determined by the application design) which can be now easily tailored on specific mining techniques. Furthermore, this specification is accompanied by the parameters that allow the instantiation of dynamic pages, the identifier of the unit of information in pages, the structure of the Web site recorded as a further specification of the typology of unit (e.g., content unit or index unit) and last but not least, the identifier of the user crawling session which allows to determine the main relevant activities performed on the application Web site by the users.

In the rest of this section, we shortly illustrate the main features of the adopted model, WebML (Web Modeling Language) [2,3], and of the rich logs that WebML-based applications produce.

2.1 WebML and WebRatio

WebML (Web Modeling Language) is a visual language for specifying the content structure of a Web application, as well as the organization and presentation of such a content in a hypertext [2,3]. It mainly consists of the Data Model and the Hypertext Model.

The *WebML Data Model* adopts the Entity-Relationship primitives for representing the organization of the application data.

The *WebML Hypertext Model* allows the description of how data, specified in the data model, are published through elementary units, called *content units*, whose composition makes up *pages*; it also specifies how content units and pages are interconnected to constitute *site views*, i.e., the application front-end. The WebML Hypertext Model includes:

The WebML Hypertext Model includes:

- The *composition model*, concerning the definition of pages and their internal organization in terms of elementary pieces of publishable content, called *content units*. Content units offer alternative ways of arranging data dynamically extracted from entities and relationships of the data schema.
- The *navigation model*, describing links between pages and content units, which have to be provided to facilitate information location and browsing.
- The *content management model*, which consists of a set of units for specifying operations for creating and updating content.

Besides the visual representation, WebML primitives are also provided with an XML-based representation, suitable to specify those properties that would not be conveniently expressed by a graphical notation.

WebRatio is the CASE tool supporting the WebML-based development (`http://www.webratio.com`). The core of WebRatio is a code generator, based on XML and XSL technologies, which is able to process WebML conceptual schemas, by translating their visual specification into concrete page templates, and generate automatically the application code. The resulting Web applications feature a three-tier architecture, in which all the relevant dimensions of a dynamic application are covered: data extraction from the data sources, code for managing the business logic, and page templates for the automatic generation of the front-end.

3 Mining Conceptual Logs

Conceptual logs are standard log files integrated with information available from the Application Server logs, from WebML application runtime, and of conceptual schema of the underlying Web application. In this Section we describe the typology of information contained in these Web logs. We also present the KDD scenarios for this specific application domain, i.e., the sequences of queries in a constraint-based mining language (`MINE RULE`) which allowed us to obtained useful, interesting and actionable patterns for Web administrators, Web application designers and application analysts.

3.1 DEI Web Application Conceptual Logs

The Web site of DEI (Department of Electronic and Information) collects one fourth of the overall clickstream directed to Politecnico di Milano, Italy. We collected the Web logs of the first consecutive 3 months in 2003. The original Web log stored by the Web server (`Apache http server`), was 60 MBytes large and is constituted by a relation that has the following information:

RequestID: the identifier of the request made by the user of a Web page;
IPcaller: IP address from which the request is originated; very often it is a proxy IP, that masks the real identification of the user.
***Date*:** date of the request,
***TS*:** time stamp of the request,
***Operation*:** the kind of operation request (for instance, `get` or `put`)
***Page URL*:** URL of the page to be transfered as a consequence of the request,
***Protocol*:** transfer protocol used (such as `TCP/IP`),
***Return Code*:** code returned by the Web server to the user,
***Dimension*:** dimension in bytes of the page,
***Browser*:** name of the browser from which the request is originated,
***OS Type*:** type of the Operating System.

To this main, standard information collected by Web server, WebML and WebRatio design applications add other information.

***Jsession*:** identifier of the user crawling session that spams over the single page requests. User crawling sessions are identified by an enabled Java browser by the Java thread identifier of a Web crawling.
***Page*:** identifier of the page generated by the application server. Very often a page is generated dynamically but this identifier is always the same for each page.
***UnitID*:** identifier of an atomic piece of information contained in a page. This identifier gives information on the type of content of a page.
***OID*:** identifier of an object (for instance, a professor, a course, a pubblication) whose content is shown in a page. This object identifier is used by the application server to instantiate in different ways dynamic pages according to the object itself. For instance, all professor pages obey to the same template that shows personal data, photo, description of the curriculum vitae of the person and of its research area. Instead, the real information that is shown for each person changes accordingly to the professor, and therefore to the `OID` parameter that identifies the person.
***Order*:** ordering number in which content units are presented in the page.

The Web Log contained almost 353 thousands user sessions, constituted each by an average of 12 page requests, for a total of more than 4.2 millions of page requests. The total number of pages (dynamic, instantiated by means of OIDs) was 38554.

3.2 MINE RULE

`MINE RULE` is an SQL-like operator for mining association rules in relational databases. A detailed description can be found in [20]. This operator extracts a set of association rules from the database and stores them back in the database in a separate relation.

Let us explain the operator with the aid of a simple example on `WebLogTable`, containing the information of the conceptual log described in Section 3.1. The following `MINE RULE` statement extracts rules that aim to provide a description of the situations that generate frequently an error in the Web server (a favorite situation for attacks). *WebLogTable* has been grouped by *RequestId*; requested rules associate values of $\langle Operation, Browser, PageURL\rangle$ with values of *Return Code*. Selected rules will have a value of returned code corresponding to an error (`WHERE` clause). Rules will have a support and a confidence greater than the minimum requested values (respectively 0.2 and 0.4).

```
MINE RULE SituationsRuturnCodes AS
SELECT DISTINCT 1..n Operation, Browser, Page Url AS BODY,
                1..n Return Code AS HEAD, SUPPORT, CONFIDENCE
    WHERE HEAD.Return Code LIKE '%error%'
FROM WebLogTable
GROUP BY RequestId
EXTRACTING RULES WITH SUPPORT:0.2, CONFIDENCE:0.4
```

This statement extracts each rule as an association of attribute values occurring within single tuples. In other statement cases, rule elements are constituted by values of the same attribute (e.g., `Page URL`) occurring in different tuples (e.g., requests) of the same group (e.g., date).

The main features of `MINE RULE` are:

- *Selection of the relevant set of data* for a data mining process. This feature is applied at different granularity levels (row level or at the group level, with the *group condition*).
- Definition of the *structure of the rules* (single or multi-dimensional association rules) and cardinality of the rule body and head.
- Definition of *constraints applied at different granularity levels.* Constraints belong to different categories: constraints applied at the rule level (*mining conditions* instantiated by a `WHERE` clause), constraints applied at a group level (instantiated by a `HAVING` predicate) and constraints applied at the *cluster* level (*cluster conditions*). For lack of space we will not make use of cluster condition in this paper.
- Definition of *rule evaluation measures.* Practically, the language allows to define support and confidence thresholds.[1] Support of a rule is computed on the total number of groups in which it occurs and satisfies the given constraints. Confidence is analogously computed (ratio between the rule support and the support of the body satisfying the given constraints).

3.3 Analysis of Web Logs with MINE RULE

We have imported into a relational DBMS (`mysql`) conceptual logs obtaining a table named `WebLogTable`. In this Section we describe in detail the KDD

[1] Theoretically, also other measures, based on body and head support, could be used.

scenarios, composed of a sequence of pre-processing, mining and post-processing queries that we have designed for discovery of useful patterns in the Web logs. These queries can be conceived as a sort of *template* that can be used to gather descriptive patterns from Web logs, useful to solve some frequent, specific or critical situations.

Analysis of Users that Visit the Same Pages. This analysis aims at discovering Web communities of users on the basis of the pages that they frequently visited.

Pre-processing Query. The mining request could be preceded by a pre-processing query selecting only those page requests that occurred frequently (above a certain threshold) thus allowing to neglet the rare page requests.

Mining Query. This query finds the associations between sets of users (IP addresses) that have all visited a certain number of same pages. In particular this number of pages is given in terms of support of the rules. (In this example, support is computed over the requested pages, since grouping is made according to the requested pages). It is an open issue whether the discovered regularities among IP addresses occur because these IP addresses have been commonly used by the same users in their pages crawling. Indeed, this phenomenon could put in evidence the existence of different IP addresses dynamically assigned to the same users.

```
MINE RULE UsersSamePages AS
SELECT DISTINCT 1..n IPcaller AS BODY, 1..n IPcaller AS HEAD,
                                             SUPPORT, CONFIDENCE
FROM WebLogTable
GROUP BY Page Url
EXTRACTING RULES WITH SUPPORT:0.001, CONFIDENCE:0.4
```

In order to instantiate in a meaningful way the preceding query, we had to perform some exploratory analysis of the source table, which is necessary to derive some meaningful values for the support threshold. Indeed, since the number of groups in which the source table is partitioned is decided by the grouping clause in the specific `MINE RULE` query, we had to launch a standard SQL query to derive the total number of page Urls contained in the `WebLogTable`. This number was 38554. Therefore, the minimum support threshold must be higher than 1/38554, otherwise every possible sets of user's IP addresses that accidentally requested one single common page would be recovered by the system. With the value of 0.001 for the minimum support we extracted 421 rules which decreases to 151 with a value of minimum confidence equal to 0.4. As a further work it would be interesting to extract some condensed representation of the set of frequent rules, such as a set of non redundant rules as proposed in [29,22].

Examples of some obtained results. In practice, in our experiments we discovered that the most frequently co-occurring IP addresses belong to Web crawlers engines or big entities, such as universities.

A similar query would occur if we wish to discover user communities which share the same user profile in terms of usage of the network resources. In this case, we would add constraints (in the mining condition, for instance) on the volume of the data transferred as a consequence of a user request. In this case examples of discovered patterns are the requests of frequent download of materials for courses, or documentation provided in the users' home pages.

Post-processing Query. As a post-processing query instead we could be interested in finding those pages that have been all visited most frequently by certain sets of users. This is a query that crosses-over extracted patterns and original data. With this request we could also discard from the discovered patterns those ones belonging to Web crawlers engines.

These post-processing queries are possible because `MINE RULE` system stores the discovered rules in the database. The main table, contains the rules, in terms of identifiers of body and head itemsets and of the rules statistical measures (which, in the current implementation are simply support and confidence). The secondary table, contains the details of the elements of rule body and head itemsets (in terms of the body and head schemas specified in the `SELECT` clause of the query).

The following two query scenarios aim at performing Web structure mining.

Most Frequent Crawling Paths

Pre-processing Query. As in previous case, the mining request can be preceded by a pre-processing selecting only those page requests that occurred frequently.

Mining Query. This query returns sequences of pages (ordered by date of visit) frequently visited.

```
MINE RULE FreqSeqPages AS
SELECT DISTINCT 1..2 Page Url AS BODY, 1..1 Page Url AS HEAD,
                                  SUPPORT, CONFIDENCE
WHERE BODY.Date = HEAD.Date and BODY.TS < HEAD.TS
FROM WebLogTable
GROUP BY IPcaller
EXTRACTING RULES WITH SUPPORT:0.002, CONFIDENCE:0.4
```

You can notice that in this case we grouped by user (`IPcaller`) and searched for sets of pages frequently occurring in the visits of a sufficient number of users (support). Notice also that we used a mining condition to constrain the temporal ordering between pages in antecedent and consequent of rules, thus ensuring the discovery of sequential patterns. We limited the search to ordered sets pages requestes in the same day by the same user IP, temporally ordered (see condition on time stamps). We also counted the total number of distinct groups in order to evaluate a meaningful value for the support threshold and we obtained that it was equal to 406. Thus minimum threshold was setted higher than $1/406 = 0.002$. With this value we obtained 7415 rules which reduced to 1773 setting the confidence threshold.

Examples of some obtained results. In practice, examples of resulting interesting patterns showed that requests of a research center page, or research expertise area were later followed by the home page of a professor. We interpreted this as a hint to the fact that people preferred to reach the personal pages by mean of a secondary access point (the research center or reserach area index) instead of the more direct index to the personal home pages. This was perhaps a sign that the general index of the global institution on people home pages was too slow inducing requests to be preferentially directed to other more little and more efficient indices.

Post-processing Query. A post-processing query can discover the sets of IP addresses originating the frequently occurring page requests. Again, this query crosses over patterns and data. We discovered by this query some publicly available IPs, of some well-known internet providers in Italy.

Units that Occur Frequently Inside Users Crawling Sessions. One of the main advantages gained by the conceptual web logs is the knowledge of the information content of the pages. These content units can give us a more precise information on the reasons why certain pages are frequently requested by the users. The following query extracts associations between two sets of content units that appeared together in at least a certain number of crawling sessions.

```
MINE RULE UnitsSessions AS
SELECT DISTINCT 1..n UnitID AS BODY, 1..n UnitID AS HEAD,
                                     SUPPORT, CONFIDENCE
FROM WebLogTable
GROUP BY Jsession
EXTRACTING RULES WITH SUPPORT:0.05, CONFIDENCE:0.4
```

Examples of some obtained results. With this query we discovered that the units that most frequently co-occurred in visits are the structural components of the Web site, such as indexes, overview page of the personnel, map pages of the site, and so on. The results of this query could be used by the Web application designers to restructure the site in order to make easier the search for those units that frequently co-occur in user visits.

Anomalies Detection. This query tries to determine the associations between pages and users that caused a bad authentication error when making access to those pages. Therefore, this query wants to determine those pages that could be effectively used by callers as a way to enter illegally into the information system.

Pre-processing Query. The mining request was preceded by a pre-processing query selecting only those page requests that occurred a sufficient number of times. This discards those requests that have been mistakenly submitted by the user (a wrongly typed password), that if not repeated many times, cannot be considered an intrusion attempt.

```
MINE RULE BadAuthentication AS
SELECT DISTINCT 1..1 IPcaller AS BODY, 1..n Page Url AS HEAD,
                                      SUPPORT, CONFIDENCE
WHERE BODY.IPcaller = HEAD.IPcaller
FROM WebLogTable WHERE Return Code='bad authentication'
GROUP BY Date
EXTRACTING RULES WITH SUPPORT:0.03, CONFIDENCE:0.4
```

You can notice that `WHERE Return Code='bad authentication'` is effectively a pre-processing operation that selects only the portion of interest of Web logs. In this query we grouped source data by date, thus identifying patterns (association of users to page requests) that are frequent in time. Notice that mining condition `WHERE BODY.IPcaller = HEAD.IPcaller` ensures that page requests (head) effectively were originated by the callers associated to them (body).

The total number of obtained rules was 80 which decreased to 72 with the confidence threshold.

Examples of some obtained results. Some examples of retrieved patterns are provided by those attempts of change of passwords, or downloading of some reserved information.

High Traffic Users

Pre-processing query. Similarily to previous queries, also this data mining query could be preceded by a pre-processing step, selecting only the frequent page requests. Indeed, rare page requests can be neglected.

Mining query. This query returns the associations between two sets of user IP addresses from which a request of pages is characterized by a large volume of data. This constraint is enforced by means of a preprocessing predicate `WHERE dimension>=1024` that selects only those requests generating high volume of traffic on the network.

```
MINE RULE HighTrafficUsers AS
SELECT DISTINCT 1..n IPcaller AS BODY, 1..n IPcaller AS HEAD,
                                          SUPPORT, CONFIDENCE
FROM WebLogTable
     WHERE dimension>=1024
GROUP BY date
EXTRACTING RULES WITH SUPPORT:0.03, CONFIDENCE:0.4
```

Notice that we grouped the input relation by date thus identifying the users that request pages characterized by a high volume frequently in time.

Post-processing query. A cross-over query can discover those pages originating the frequently occurring page requests.

Examples of some obtained results. As examples of discovered patterns there are the requests of frequent download of materials for courses, or documentation provided in user home pages.

Errors Correlated to Usage of an Operating System. This query returns associations between the operating system and the error code frequently returned by the Web server.

```
MINE RULE OSErrors AS
SELECT DISTINCT 1..1 OStype AS BODY, 1..n Return Code AS HEAD,
                                             SUPPORT, CONFIDENCE
WHERE BODY.OStype=HEAD.OStype
FROM WebLogTable  WHERE Return Code LIKE '%error%'
GROUP BY Date
EXTRACTING RULES WITH SUPPORT:0.01, CONFIDENCE:0.4
```

Notice the pre-processing predicate (`WHERE Return Code ..`) that selects only the page requests that result in some errors. The total number of retrieved rules was 296. This query is similar to query named `BadAuthentication` for the discovery of anomalies and can be useful to test the reliability and robustness of a new Web application.

Users that Visit Frequently Certain Pages. This request aims at discovering if recurrent requests of a set of pages from a certain IP exist. This puts in evidence the *fidelity* of the users to the service provided by the Web site.

```
MINE RULE UsersPages AS
SELECT DISTINCT 1..1 ipaddress AS BODY, 1..2 Page Url AS HEAD,
                                         SUPPORT, CONFIDENCE
WHERE BODY.ipaddress = HEAD.ipaddress
FROM WebLogTable
GROUP BY RequestId
EXTRACTING RULES WITH SUPPORT:0.001, CONFIDENCE:0.4
```

As in previous experiments we setted the minimum support threshold after a prior inspection of the total number of received requests (which was equal to 4.2 millions). However, the mining condition reduced the number of effective groups from which a valid association rule was present. Also, the support threshold helped to reduce the volume of the result which finally contained only 421 rules.

Examples of some obtained results. Examples of patterns we discovered are provided by the pages that allow the download of material, such as course slides and research papers published publicly in the personal home pages. Other similar queries, on content units (instead of pages) are also a useful suggestion and allow to acquire a lower granularity in discovering the user crawling paths. Patterns resulting from this request confirm the previous results (the most recurrent requests are download of materials from the Web site). This observation gave to system administrators useful informations to manage the bandwidth of the network in more optimized ways.

3.4 Query Execution Times

Figure 1 reports for completeness the execution times of queries in the experiments on Web log. You can observe one bar representing the execution time of each component of the system in execution (parser, optimizer of the execution plan, pre-processing phase, data mining itemset and rules extraction phases). With this experiment we can also compare the relative impact on execution times of the various components. In another Chapter of this book we discussed in detail the algorithms and the data structures adopted by the `MINE RULE` system for executing some of the queries. In particular, that Chapter can be consulted to obtain more information on the particular strategy adopted to execute the queries when mining conditions are boolean expressions of terms in the form `[BODY|HEAD].<Attribute> <ComparisonOperator> <Attribute-Value>` and no clustering condition is present. That Chapter discusses mainly on the opportunity to develop a constraint incremental evaluation strategy exploiting previous queries results, stored in the database. It compares this incremental strategy

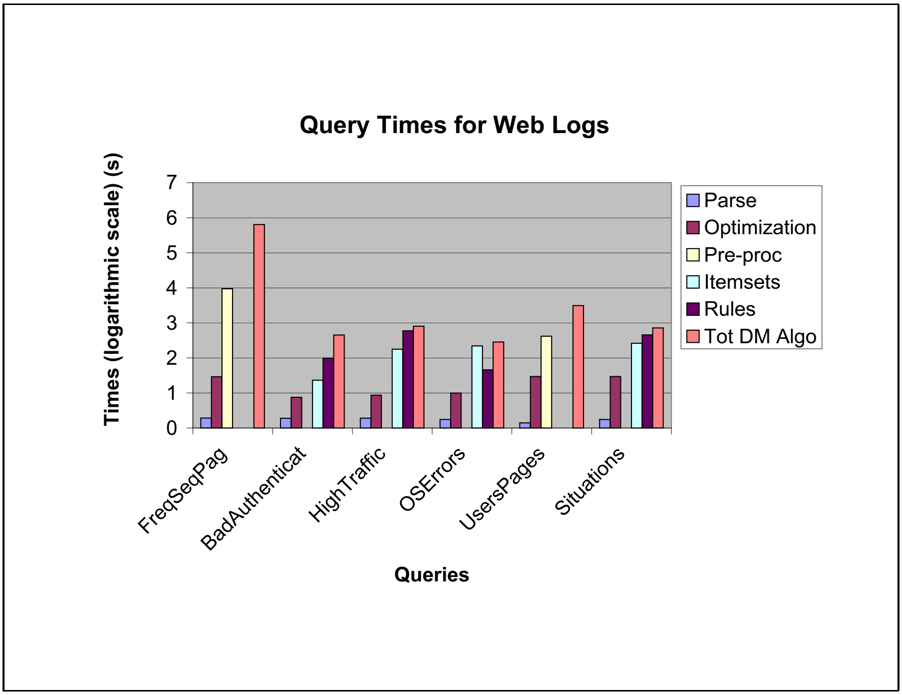

Fig. 1. Query Execution Times in Experiments on Web Logs

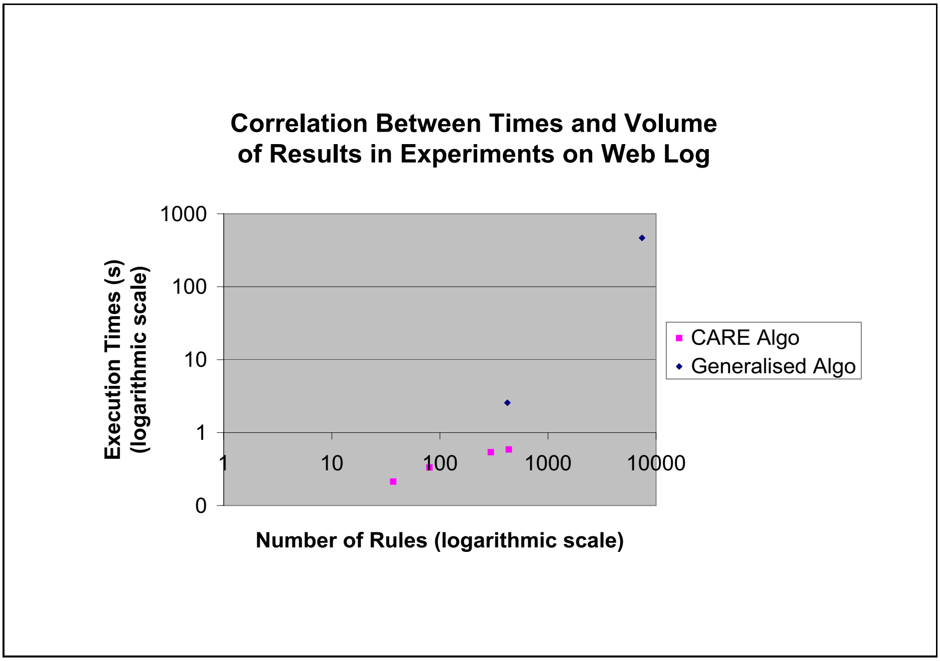

Fig. 2. Correlation Between Times and Volume of Results in Experiments on Web Log

with a strategy that works from scratch (the algorithm is called *CARE*). Still, in this Chapter we present only queries executed from scratch. Indeed, since the various scenarios have just the purpose to address typical domain-specific situations they are un-correlated (one with respect to the others). Therefore, they cannot be solved with the proposed incremental strategy.

[1] can be consulted to obtain details on the execution algorithm that addresses the general cases of a `MINE RULE` statement (called generalised algorithm). In fact, `MINE RULE` can be instantiated in very different queries, comprising different constraint typologies and even retrieving multi-dimensional rules. For instance, the generalised algorithm takes care when (i) body or head schemas are different or defined on a list of attributes; when (ii) "cross" mining conditions are present (i.e., conditions between `BODY` and `HEAD`, under the form `BODY.<Attribute> <ComparisonOperator> HEAD.<Attribute>`); (iii) conditions on clusters or aggregate functions are specified. Therefore, a rich set of different algorithms has been implemented in the system in order to better exploit the specificity of each query (in terms of constraints typologies and properties, regularities in the selected source data, such as functional dependencies between the attributes in the rules and in the mining condition).

Moreover, in Figure 2 it is possible to observe the diagram showing the correlation between query execution times and number of rules in the results (scales are logarithmic). You can immediately observe that they are approximately linearly correlated. However, the trend in execution times of the generalised algorithm is much more severe, because of course it addresses a general and more complex problem at the expenses of efficiency.

4 Application 2: Financial Data Analysis

4.1 Dow Jones Stocks Exchange Index

Dow and Jones, two economists of the XX century, with a set of articles appeared in 1900-02 in Wall Street Journal, defined a set of few stocks whose value could have been used with the purposes of monitoring the evolution of the USA economy. Initially stocks were grouped in two sets: transportation companies and industrial companies (energy production, mineral extraction, and so on). Nowadays, the index named *Dow Jones 30* contains 30 stocks of companies still strictly connected to production activities in USA, such as Microsoft, Intel, AT&T, General Motors, Mc Donalds', etc... and is still used as a meter of judgement on the evolution and wealth of the american economy because it is grounded on some big companies whose core activity is the production of consumers' goods or services. However, it is very much dynamic and in a temporal interval of few years it can change a significant part of its constituting stocks.

In Figure 3 we report the daily percentage variation of Dow Jones index from 1896 till 2003.

4.2 Technical Analysis

As opposed to fundamental analysis, which is based on the study of the corporations' activity under a macro economic view [10], technical analysis is based on the a-posteriori study of the stocks quote trend. Technical analysis [24,13,15,8,24] has been founded at the beginning of the XX century by some economists such as Dow, Hamilton and Rhea.

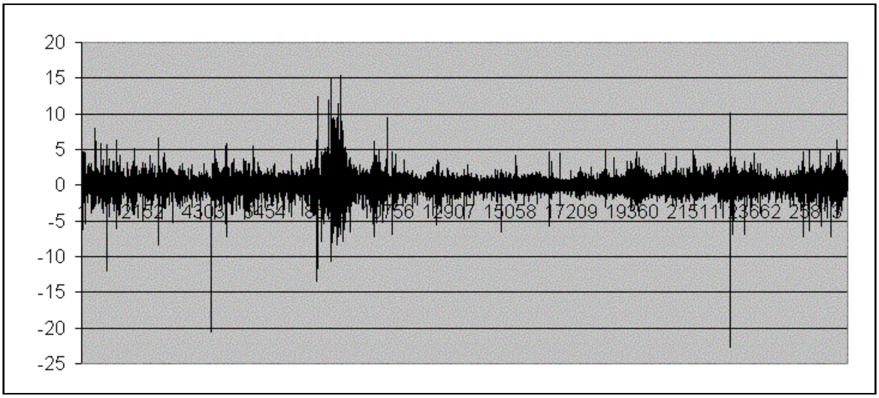

Fig. 3. Dow Jones index percentual daily change from 1896 to 2003

Some Foundation Hypothesis of Technical Analysis

- Quote is considered as the synthesis of the investors' confidence in the stock intrinsic value [18];
- investors (psycological) reactions to certain events are always the same and will repeat in the future [16];
- current trend in quotes will continue until some event makes it change. This is useful for the detection of patterns that determine the inversion of the quote trend.

Technical analysis is based on the feeling that the quote of a stock is based on the investors' faith in that stock value at the moment, and this determines that quotes have a cyclic evolution, able to reach a maximum value or a minimum value in a certain period of time (named as *resistance* and *support* respectively). The ability to determine these values is very important for the stock market analyzers because helps investors to determine the best time to sell or buy the stocks [19].

Another important event that analyzers try to determine is the point in time when the quotes of a stock change their trend (from positive to negative or viceversa). Indeed, for an investor who wishes to sell his/her stocks, the best time to sell them is the point in which the trend from positive becomes negative. At this point, the stock reached its maximum value and afterwards it will start to decrease its value. Therefore, if the investor sells in this point in time he/she will be able to make the best profit from the sold. Analogously for the purchase: the best point in time to buy a stock is when the trend from negative turns into positive. At this time the stock value has reached the minimum value and from this time on, it will start to increase making the customer spend more for the same stocks.

4.3 Japanese Candle-Sticks

Candle-sticks have been originally proposed by Japanese market analyzers to study the rice market. A single candle-stick represents the synthesis of the exchanged stocks occurred in one period of time (such as a date or a week) for a given stock. Graphically it is similar to box-plot used for exploratory and descriptive data analysis: it is constituted by a box located in a `time x quote` dimension plot, whose horizontal borders identify the open and close value of the stock in that time period. A candle-stick is colored as follows.

Black candle-stick: represents a time period in which the open value is higher than the close value. This identifies a period in which the stock lost part of its value.

White candle-stick: represents a time period in which the open value is lower than the close value. This identifies a period in which the stock gained part of its value.

Two vertical lines might depart from the box borders: the lower line represents the minimum value reached in the period while the higher the maximum. If some

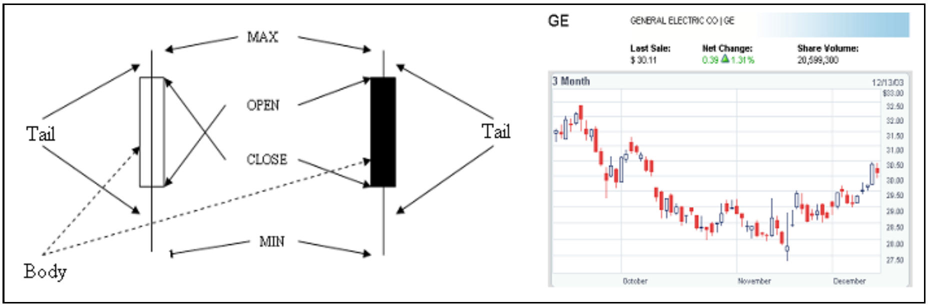

Fig. 4. Two candle-sticks and their usage in daily stock quotes plots

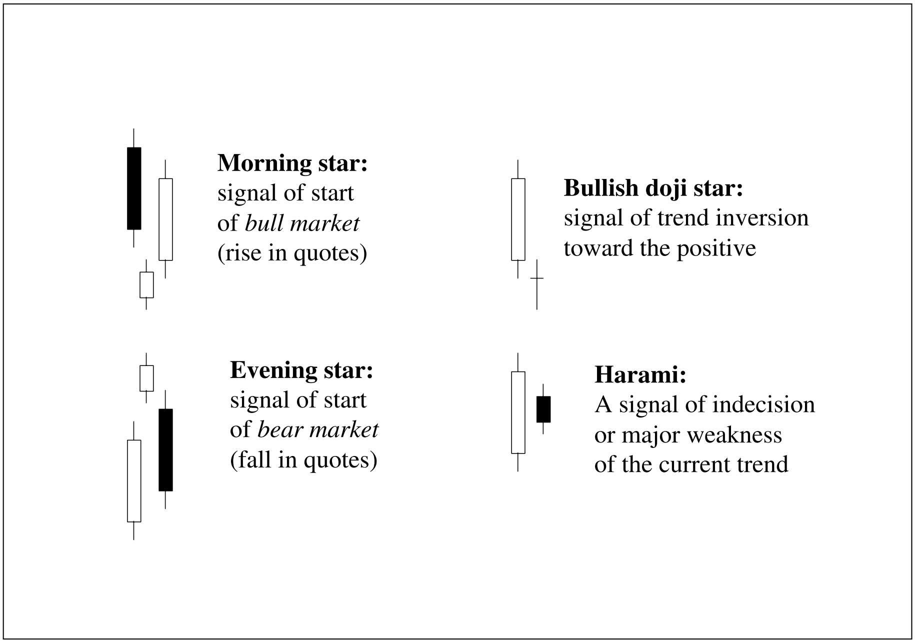

Fig. 5. Some representative candle-sticks and their predictive meaning

of these lines are absent it means that the minimum or the maximum value is reached with the open or close value (Figure 4 on the left).

The right part of Figure 4 shows an example of box-plots positioned in a `time x quote` graph. This offers an immediate and visual representation of the evolution of the value of a stock in the time, with candle-sticks providing the quality (positive or negative) of the stock exchange period.

For a long time, technical analysts have used candle-sticks to grasp with a visual representation the evolution of quotes in time. They have elaborated many configurations of the different candle-sticks in time, some of which are reported in Figure 5 with a brief explanation of their meaning. Some technical analysts considered these patterns as an “alarm” able to predict in conjunction to other observational elements (such as the volume of exchanges) the immediate evolution of stocks quotes. This is based on the principle that the quote of a stock is determined by the sell/buy law by investors themselves. Indeed, their willingness to buy or sell is determined by the market reactions to certain events which tend to repeat in the future. Therefore, the quote of a stock represents the “summa” of the willingness of the investors to buy the stocks and on their confidence in the stock intrinsic value.

4.4 Analysis of Dow Jones 30 Stocks Exchange with MINE RULE

We downloaded data on the daily market exchanges of stocks in Dow Jones 30, from 1997 till 2002, transformed it in relational form and imported in a DBMS (mysql). Since the index is very dynamic and some stocks were changed during this period, some of the analysis were performed on a subset of this time period (such as one or two years, only). The data set we collected is 2.5 MB large, and contains the following information for every date:

date: the date of the trade exchange;
ticker: the symbol identifying the stock;
open: the opening value of the stock;
close: the closing value of the stock;
min: the minimum value of the stock;
max: the maximum value of the stock;
volume: the total number of exchanged stocks of a ticker in the date.

In the following we report the KDD scenarios we developed for this financial application domain.

Frequent Candle-Stick Sequences. The following statement returns the pairs of candle-sticks that have been found in two successive dates by a relevant number of different stocks.

```
MINE RULE frequent-candle-sticks-sequences AS
SELECT DISTINCT 1..1 candle-stick AS BODY,
                1..1 candle-stick AS HEAD, SUPPORT, CONFIDENCE
WHERE BODY.date=HEAD.date-1
FROM dj30quotes
GROUP BY ticker
EXTRACTING RULES WITH SUPPORT:0.30, CONFIDENCE:0.40
```

As you will be able to see in Section 4.5 in which the execution times of the queries are reported, this query had one of the worst performance (though the number of retrieved rules with these support and confidence thresholds is low: only 78). However, notice the mining condition is checking all the possible consecutive dates in the temporal interval of 5 years, which is a quite large domain.

Examples of some obtained results. We launched this statement on the stock quotes of different years, separately, and compared the results. We noticed for example that in years 1997, 1998 and 1999, the following candle-stick sequence, `a black candle-stick immediately followed by a white one`, has been found much more frequently than in later years, 2000, 2001 and 2002 (between 23% and 76% of all the stocks). In fact, in the first years investors obtained high profits (Dow Jones index roughly doubled its value) and fostered good faith in the market; on the contrary, in later years, great losses were experienced. We believe that this sequence can be interpreted as a signal of trading which is going to be soon saturated (intuitively, that everyone is willing to buy). However, in years 2000-2002, we found that sequences made by `three consecutive candle-sticks including only black candle-sticks and white candle-sticks with both the tails` (i.e., such as that the minimum and maximum laid outside of the body of the candle-stick) were surprisingly much more frequent than in previous years (their frequency was almost doubled). Our interpretation to this is that this type of candle-stick sequence might be a signal of a "nervous" market, which is a sign of indecision and might constitute a suggestion of refrain from a new investment.

Pairs of Stocks with Similar Behavior. This statement searches for the stocks with a similar behavior in time. The similarity of behavior is decided according to the common candle-sticks types the two stocks show in the same dates.

```
MINE RULE similar-tickers AS
SELECT DISTINCT 1..1 ticker AS BODY, 1..1 ticker AS HEAD,
                                     SUPPORT, CONFIDENCE
WHERE BODY.candle-stick = HEAD.candle-stick
FROM dj30quotes
GROUP BY date
EXTRACTING RULES WITH SUPPORT:0.30, CONFIDENCE:0.40
```

Examples of some obtained results. In Figure 6 we show the plot of quotes in 2002 of the pair of stocks in the result that are most similar: `Hewlett-Packard` and `Microsoft Corp`. You can notice actually how much they are similar. Another set of very similar tickers is composed by `Home Depot Inc.`, `Walt Disney-Disney C.`, `JP Morgan Chase Co.` and `American Express Inc`.

Verification of Price Percentage Variation by Volumes. The main aim of the following queries is the verification of one of the most well-known principles in stock trading [18]: increasing volumes in the exchanges of a stock is a signal of a broader participation among investors; that is, contextually to increasing volumes one could expect a great movement in prices.

Pre-processing query. We preceded the real mining query with a pre-processing query with the purpose to identify the high volumes. This query computes for

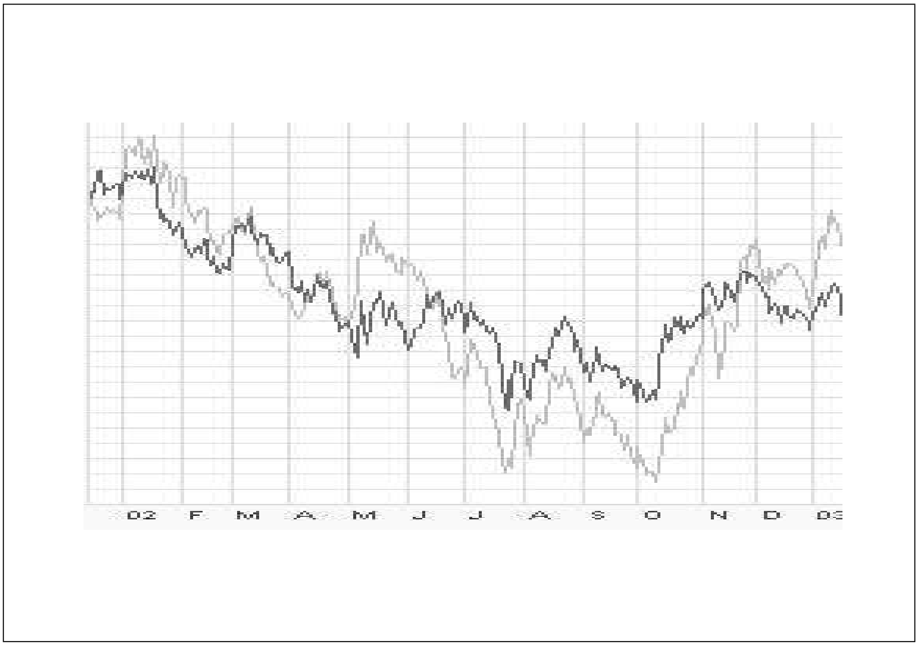

Fig. 6. Comparison between plot of Hewlett-Packard and Microsoft stock daily quotes in 2002

each stock and each date the ratio r between the exchanged volume (i.e. the number of exchanged stocks) in the day and the average volume exchanged in the whole year. In this way we could label each stock exchange date with a boolean attribute called `high_volume`.

We considered for the identification of high volume dates, for any single stock, different values of the ratio r. In this example, we experimented a value of $r = 300\%$ uniform for all the stocks and selected in source relation only the stock exchanges characterized by a high volume.

Furthermore, with the aid of another query, we found also the percentage change in quotes during the day computed with respect to the close price of the previous day, called `varp`, defined as follows:

$$varp = \frac{(close - open)}{open}$$

Analogously to what is done with high volumes, we are interested in high variations in prices, and labeled each date with `high_varp` if

$$varp >= 5\% \vee varp <= -5\%$$

Both `high_volume` and `high_varp` will be used in the following mining query.

Mining query. This statement searches for frequent associations (in time, i.e., occurred in a large number of weeks of the year) between a high volume and a high percentage variation of price of the same stock in the same date.

```
MINE RULE VarpByVols AS
SELECT DISTINCT 1..1 high_volume AS BODY, 1..1 high_varp AS HEAD,
                                                    SUPPORT, CONFIDENCE
WHERE BODY.ticker=HEAD.ticker AND BODY.date=HEAD.date
FROM dj30quotes
GROUP BY week
EXTRACTING RULES WITH SUPPORT:0.001, CONFIDENCE:0.001
```

Examples of some obtained results. From the obtained results, we can say that 86% of the exchanges in which the price percentage variation (`varp`) is above 5% or below −5% occurs with a volume greater than 125% of the daily volume, averaged on the whole year. This confirms the initial hypothesis we wanted to test.

The above query needed a certain amount of preparation tests, since we had to discover the value of the ratio r and of the percentage variation *varp* that best confirm the evidence. In Section 5 we will discuss more on the KDD process that was necessary to prepare the source data with a discretization step (obtain the boolean derived attributes `high_volume` and `high_varp`).

Stocks with White Candle-Stick in the Same Date. The following search is motivated by the necessity of identifying a set of stocks suitable to constitute the investors' portfolio.

Pre-processing query. Similarly to previous query, we performed a pre-processing query selecting the stock exchanges occurred with high volumes. (We considered again the same values used previously).

Mining query. We launched the following mining query on the stocks in 1997, the first year in the observed time period, and searched for the pair of stocks that frequently had a white candle-stick in the same dates with a relevant percentage of price variation.

```
MINE RULE white-candle-stick-pairs AS
SELECT DISTINCT 1..1 ticker AS BODY, 1..1 ticker AS HEAD,
                                     SUPPORT, CONFIDENCE
WHERE BODY.candle-stick-type='white' AND
      HEAD.candle-stick-type='white' AND
      BODY.varp>5 AND HEAD.varp>5
FROM dj30quotes
GROUP BY date
EXTRACTING RULES WITH SUPPORT:0.01,CONFIDENCE:0.40
```

Examples of some obtained results. This query returns 870 rules. The ones that show the highest support and confidence are constituted by six stocks such as UBS AG, General Electric, Honeywell, Intel, Merck & Co. and Procter & Gamble. We can notice how among all the rules, many of them are redundant. What is really meaningful in this case is the condensed representation of itemsets that occur in the same situations (dates): the concepts, in Formal Concept Analysis [28,21,22]. Further work on the MINE RULE system, which actually does not provide support for a condensed representation of itemsets, is still needed to improve the representation of itemsets in a way that is more meaningful.

Post-processing. This step was used to evaluate the portfolio composed by previously selected stocks. This portfolio was monitored for the following 4 years and outperformed Dow Jones index from 5% to 11% in any year. Furthermore, it gained in each single year from the 5% to 29% of the total investment. This is a very useful result and is a first demonstration of the practical usefulness of these techniques.

Discovery of Frequent Doji Star Candle-Sticks in Time. The following statement searches for the pairs of successive dates in which most of the stocks show a *doji star candle-stick*. Specialized literature on technical analysis considers this pattern as a signal of reversal of trend followed by a signal of indecision of the market. It can be viewed as an alarm signal. Indeed, we discovered this pattern in spring and in autumn of 2002. An example of this pattern can be observed in Figure 5 under the name of *bullish doji star* that is very similar to the doji star pattern with the exception that the first candle-stick is white instead of black. In the following MINE RULE, you can observe the mining condition

$$HEAD.open + 0.003 * HEAD.open > HEAD.close\ AND$$
$$HEAD.open - 0.003 * HEAD.open < HEAD.close$$

which serves to search for the cross pattern with a tolerance between the open and the close value of a 0.3%. In fact, a perfect match would be very unprobable. Notice, how this tolerance in the comparison between open and close values plays a similar role to soft comparison in fuzzy query languages, since it allows us to test a predicate under some weaker conditions, necessary in the stock financial domain in which a certain amount of noise is always present.

```
MINE RULE freq-doji-star-candle-sticks-in-time AS
SELECT DISTINCT 1..1 date AS BODY, 1..1 date AS HEAD,
                                  SUPPORT, CONFIDENCE
WHERE BODY.date=HEAD.date-1 AND
      BODY.close>BODY.open AND BODY.close<HEAD.open AND
      (HEAD.open +0.003*HEAD.open > HEAD.close AND
       HEAD.open -0.003*HEAD.open < HEAD.close)
FROM dj30quotes
```

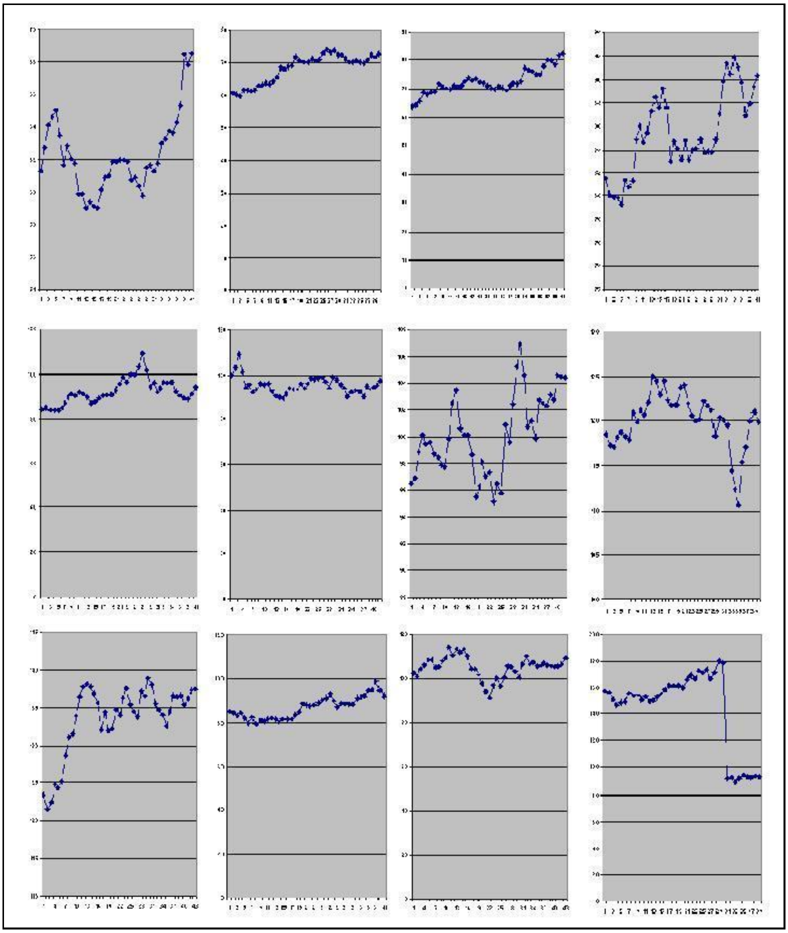

Fig. 7. Quotes plot of Microsoft stock in the time period (2 months) immediately following a bullish doji star occurrence

```
GROUP BY ticker
EXTRACTING RULES WITH SUPPORT:0.30, CONFIDENCE:0.40
```

86 total rules were discovered for all the tickers. In Figure 7 we report the stock trend immediately following the detection of a bullish doji star in `Microsoft` stock and indeed, you can notice how the trend is usually significantly positive, especially in the first part of the evolution. A bullish doji star is the opposite: it corresponds to a signal of indecision followed by a reversal in the market trend. We also checked for the presence of this pattern in a stock, like `Microsoft Corp.`, that is the stock characterized by the highest capitalization in the market.

Discovery of Morning Star Candle-Sticks. The following statements search for the dates in which most of the stocks show a *morning star* candle-stick pattern. As you might recall, this pattern is composed by three candle-sticks, so that we need in this case two separate MINE RULE statements: the first one, called 1st-part-morning-star searches for the first part of the pattern (first two candles); the second one, 2nd-part-morning-star is needed for the second part (second and last candle). Finally, a simple SQL query joins the results looking for the complete pattern where the first part of the pattern is followed by the second part for the same stock and in an immediately successive date. Of course, the intermediate candle must be the same in the two parts of the pattern. Of course this methodology could be extended for patterns of arbitrary length and thus to the discovery of sequential patterns.

```
MINE RULE 1st-part-morning-star AS
SELECT DISTINCT 1..1 date AS BODY, 1..1 date AS HEAD,
                                  SUPPORT, CONFIDENCE
WHERE BODY.date=HEAD.date-1 AND BODY.close<BODY.open AND
      BODY.close>HEAD.close AND HEAD.close>HEAD.open
FROM dj30quotes
GROUP BY ticker
EXTRACTING RULES WITH SUPPORT:0.30, CONFIDENCE:0.40

MINE RULE 2nd-part-morning-star AS
SELECT DISTINCT 1..1 date AS BODY, 1..1 date AS HEAD,
                                  SUPPORT, CONFIDENCE
WHERE BODY.date=HEAD.date-1 AND BODY.close>BODY.open AND
      BODY.close<HEAD.open AND HEAD.close>HEAD.open
FROM dj30quotes
GROUP BY ticker
EXTRACTING RULES WITH SUPPORT:0.30, CONFIDENCE:0.40
```

90 and 131 occurrences, respectively of the first and second part of the morning-star pattern, were discovered in the five years.

Post-processing. A post-processing standard SQL query performs the join between the result of the 1st-part-morning-star query and of the 2nd-part--morning-star query taking care that the head of the first part coincides with the body of the second part. This guarantees that the two parts of a morning-star candle-stick pattern are effectively found in two consecutive days. In 2002, only 29 occurrences of the complete pattern were discovered. (We tested this query only on one year because the intermediate table for the 5 years was too large to perform the join in the database with a reasonable time).

One final observation concerns how some fuzzy conditions, similar to what done for query freq-doji-star-candle-sticks-in-time could be useful to gain a certain amount of flexibility in evaluating the time interval between the occurrence of the first and the second part of the candle-stick pattern.

```
SELECT D1 ticker, D1.date
FROM dj30quotes2002 D1, dj30quotes2002 D2, dj30quotes2002 D3,
     1st-part-morning-star F, 2nd-part-morning-star S
WHERE D1.ticker=D2.ticker AND D2.ticker=D3.ticker AND
      D1.date=F.body AND D2.date=F.head AND
      S.body=D2.date AND D3.date=S.head
```

Examples of some obtained results. The results confirm that these candle-stick patterns are quite rare. (In 2002, they were present mainly in August and December, a period in which the market raised again).

4.5 Query Execution Times

Figure 8 reports for completeness the execution times of queries in the experiments on Dow Jones 30. You can notice how the execution times of the join query has evaluation times comparable to the extraction of rules by MINE RULE.

Moreover, in Figure 9 it is possible to observe the diagram showing the correlation between query execution times and the number of rules in the results (scales are logarithmic). You can immediately observe that they are clustered around a central point with the exception of some outliers. If we go to observe with more attention of which queries they consists of, we can see that the best query (first outlier for the generalised algorithm, working with a cross-condition between body and head) is similar_tickers which has a simple

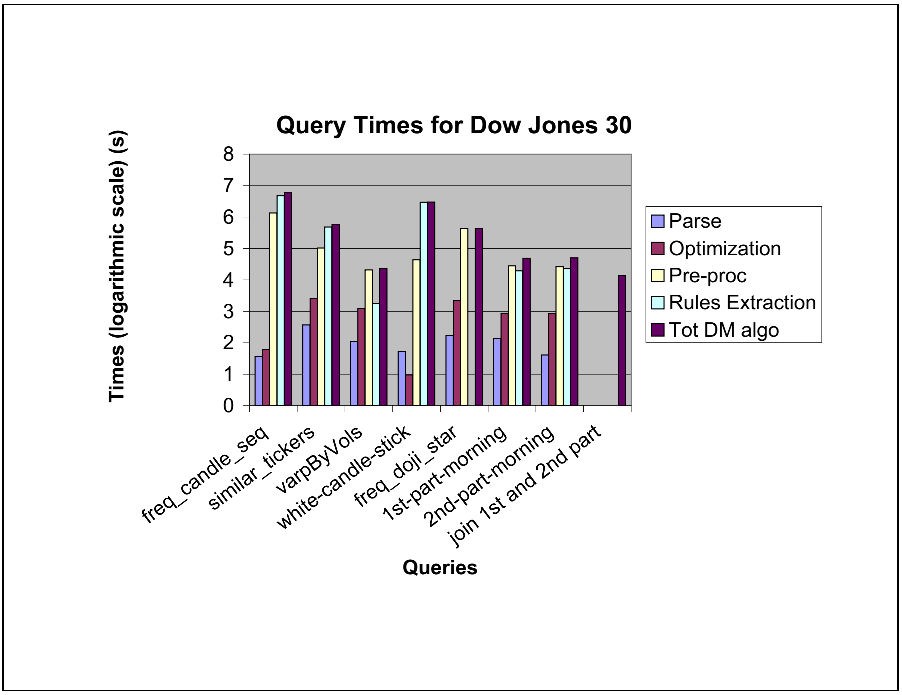

Fig. 8. Query Execution Times in Experiments on Dow Jones 30

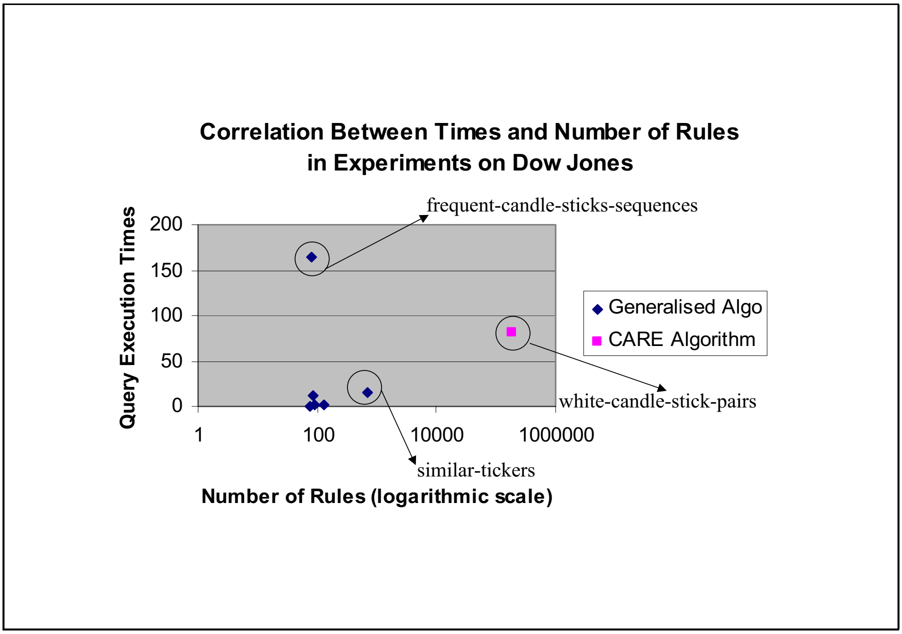

Fig. 9. Correlation Between Times and Volume of Results in Experiments on Dow Jones

mining condition (evaluated on equality on a little set of values: the 10 different candle-sticks). Instead, the worst query for the generalised algorithm is `frequent_candle_sticks_sequences`: it must evaluate the cross mining condition on the set of different trading dates, which is pretty big in a time interval of some years (4764 values). Finally you can observe again that `CARE` algorithm works much faster than the generalised algorithm. Indeed, in this experiment, `white-candle-stick-pairs` evaluated 189660 candidate rules in a lower time than the time required by the generalised algorithm to retrieve only 78 rules (but on different mining conditions). However, we must observe again, as we did already with the experiments on Web log, that this confirms the fact that this is the price to pay to have the possibility to treat more general conditions in queries. For instance, cross mining conditions (i.e., a comparison between body and head features) could be very important in practical applications.

5 Evaluation of Discovered Knowledge

In previous Section we reported the sequences of queries submitted to `MINE RULE` system in order to discover useful knowledge in practical applications, such as the analysis of Web logs for Web usage mining, and of financial data (Dow Jones, in particular). We can draw now some conclusions on the discovered patterns.

In order to be useful, discovered patterns must be:

1. interesting for the user/analyst
2. actionable, in the sense that immediate actions/decisions can be taken as a consequence of their observation.

In constrained based mining the first point is immediately fulfilled by the retrieved patterns, because by definition extracted patterns satisfy user defined constraints. Indeed, constraints are provided by the analyst to the system in order to identify the desired patterns and discard all the remaining ones. Desired patterns could be the interesting ones for many reasons. First of all because they occur frequently, and therefore they refer to a statistically relevant number of events occurred in the application domain. Secondarily because some user constraints are able to discriminate the properties of the desired pattern class with respect to some contrasting classes. However, notice that sometimes it is not easy to identify the right constraint (or at least the right constant value in a comparison). For instance, in some of the examples, as in `freq-doji-star-candle-sticks-in-time` we adopted a sort of "fuzzy" constraints. In other cases, such as in `VarpByVols`, a preparatory session was necessary. We generated derived attributes (for the ratio between daily exchange and average volume and for the price percentage variation) and discretised them in two boolean attributes (`high_volume` and `high_varp`). The discretization was tested several times (using different thresholds) and the results used later, during the mining step.

The second point is more difficult to establish in an absolute sense. Usually, a post-processing phase, following the proper mining phase is necessary to establish if the retrieved patterns are actionable. Generally, a cross-over query that retrieves the original data in which a pattern occurs is necessary to reveal in which way or which data representing real life entities are involved in the phenomena described by the patterns themselves. For instance, in the Web application domain, if patterns describing users attempts to change passwords, or putting in evidence which user configuration settings more frequently are correlated to system errors are employed to identify the users causing the problem, those patterns are immediately actionable because are self explaining. Other patterns, such as patterns on the users' most frequent crawling paths, are more difficult to translate immediately in some actions on the Web applications because might indicate a high value path or content in the Web application or non perfectly adequate tools (such as search engines or indexes) in the site, as well. In this latter case, this might involve a new design of the application, the hypertext structure and of main the content areas that organise the Web site. Finally, the discovered patterns could also identify two content areas that are contextually correlated but could not require the involvement of a new Web site design. This is the case in which it is necessary providing the users with some additional suggestions on the pages where the user could find some content in which he/she could be interested in (see recommending systems). As regards the financial domain, an example of an actionable pattern is constituted by the pairs of stocks exhibiting frequently a white candle-stick because they can be used to suggest the composition of users' portfolios. However, we should observe that in this case we discovered many rules that Another example, is the detection of a bullish doji star pattern which is a suggestion for investors to perform a sell or buy of stocks.

6 Guidelines for Using an Inductive Database in the Web Mining Application Domain

In previous Sections we have described the process we underwent for mining a Web site. This process is briefly summarised here in order to give the user an abstraction on this process, of the difficulties in which he might be involved applying an inductive database to the analysis of a Web application load, to the identification of the profile of the users in terms of frequent visits and to the usability of the Web site.

These steps are very much alike to the traditionally well-known KDD process for the discovery of knowledge from a database:

1. customization and storage of the Web log
2. preparation of the data
3. individuation and selection of the interesting data
4. mining phase
5. post-processing of the result
6. interpretation

The *customisation and storage of the Web log* corresponds to the step of loading and integration of the data that are relevant to the analysis. This step comprises the memorization of all the elements that result useful during the analysis, such as user identification, user session identification, Web crawlers robot exclusions (because they automatically spam all the Web sites and thus show a typology of interaction with the Web application that is not leaded by the real information content displayed by the pages).

Preparation of the data is a step that is often necessary for increasing the performance of the following mining phase. It consists in the selection of the data that probably will be involved in the interested patterns. Therefore this phase allows pruning of large volumes of data that will not contribute to the searched patterns. For instance, if we are looking for frequent crawling paths by users, we can immediately discard all those requests referring to pages that have been requested a little number of times. On the contrary, if we want to identify the `top k pages` in the user preferences, we should probably discard requests to those pages that, although the most frequently selected, do not provide the user the final content he is really looking for. These pages are probably some indexes and the map or overview pages of the Web site – in conclusions, those elements that structurally are needed to the crawling of the Web site itself.

Individuation and selection of the interesting data consists in definition of the constraints (and of their parameters values) that define the interesting patterns for the user and will probably later result in actionable patterns.

Some examples are provided, for instance, by the patterns that describe the profile of those users whose requests frequently provide the greatest traffic load over the net; the parameter values in this case express the volume in bytes that defines a congested traffic situation.

The mining phase consists in the execution of the mining query provided by the user and incorporating user constraints on the prepared and selected data. It results in a set of patterns that will later be post-processed in the following phase either for visualisation or for evaluation purposes.

Post-processing can consist in pattern selection, for eliminating pattern redundancies and increase the quality of the result, or further queries over both patterns and data. Cross-over queries are often necessary in order to evaluate pattern actionability. For instance, if a pattern describes the top k most frequently occurred pair of pages, we probably would also be interested in the pages themselves, in their content and discover in which way they are related to each other. Probably also an interesting issue is the profile of the users who frequently requested (at least one of) them. All these questions can be answered by some post-processing queries that do a cross-over between discovered patterns and the available data both in the Web log and in the Web site.

Interpretation phase inspects both the results of the mining phase and the results of the post-processing phase and decide how to translate in practice the results obtained by previous phases (pre-processing, mining, post-processing) and the deployment. Results of this phase consists either in actions over the Web application design or in the decision of performing further queries, starting again to execute some steps of the discovery cycle, either from the first step, the second or the third step (customisation and storage of the Web log, preparation or selection of interesting data).

7 Conclusions

We presented the application of inductive databases to two practically important case studies. The first one is the analysis of Web logs of the main Web application of a University Department. The Web log was a conceptual log, obtained by integration of standard (ECFL) Web server logs with information on Web design application and Web pages content information. The second one is the analysis of stocks quotes from the Dow Jones index, from 1998 till 2003. We adopted Japanese candle-sticks, a descriptive pattern proposed in technical analysis to determine in conjunction with other relevant events the main occurrences in time of certain relevant events in the evolution of stocks quotes.

With both these applications we applied and evaluated the usability and expressive power of a mining query language – MINE RULE. The purpose was to verify its feasiblility in practice to solve real case problems and experiment the suitability of some KDD scenarios, developed for inductive databases.

KDD scenarios have been previously produced as a set of characteristic queries solving some frequently asked questions (mining problems) from inductive database end users/analysts in order to recover from frequently occurring problems in their environment. We showed the possibility to employ these scenarios by means of the mining query languages.

The result of the queries provides us also an evaluation of the expressive power of the designed mining languages for inductive databases in `CInQ` project on the development of inductive databases (EU project IST-2000-26469), e.g., `MINE RULE`. It proved to be simple but yet a powerful query language for mining, because with the aid of few *template* queries the user is able to afford the main critical problems respectively in Web usage mining and in financial technical analysis. Indeed, this mining language is provided with expressive constraints and querying predicates (on single or aggregate properties) that are suitable to extract the description patterns needed to analyse the actual data.

In the Web domain application obtained patterns can be exploited for the definition of effective navigation and composition of hypertext elements to be adopted for improving the Web site usability. We also obtained some concrete examples of interesting or suspicious event that are useful to the end-users (Web and system administrators).

In the financial domain, obtained patterns can be effectively used to detect stock quotes evolutionary similarities, to select the stocks for the formation of the investors portfolios and to study the stock trade behavior in general.

The examples of query we provided show that the mining language is powerful, and at the same time versatile because its operational semantics seems to be the basic one. The grouping clause (that corresponds to described entities), the rule attributes selection (that corresponds to observed entities) and the support computation (that corresponds to statistical relevance of the rules) allow the users, in some of their combinations to specify completely different and new problems simply by adoption of a different choice of attributes in grouping, rules and conditions. The results of the different mining queries can then be composed with standard SQL queries (in pre- and post-processing phase) forming a sequence of queries that constitute a KDD scenario for the application of an inductive database to the particular domain analyzed. Indeed these experiments allow us to claim that Mannila and Imielinski's initial view on inductive databases [14] was correct: <<There is no such thing as real discovery, just a matter of the expressive power of the query languages>>.

References

1. M. Botta, R. Meo, and C.Malangone. Association rules extraction with mine rule operator. Technical report, RT73-2003, Dipartimento di Informatica, University of Torino, Italy, April 2003.
2. Stefano Ceri, Piero Fraternali, and Aldo Bongio. Web modeling language (webml): a modeling language for designing web sites. In *Proc. of WWW9 Conference*, May 2000.
3. Stefano Ceri, Piero Fraternali, Aldo Bongio, Marco Brambilla, Sara Comai, and Maristella Matera. *Designing Data-Intensive Web Applications.* Morgan Kaufmann, San Francisco, CA, 2002.
4. Apache Cocoon. Cocoon. http://xml.apache.org/cocoon/.

5. R. Cooley. *Web Usage Mining: Discovery and Application of Interesting Patterns from Web Data.* PhD thesis, University of Minnesota, 2000.
6. R. Cooley, P.N. Tan, and J. Srivastava. *Discovery of Interesting Usage Patterns from Web Data.* LNCS/LNAI. Springer Verlag, 2000.
7. G. Das, K.-I. Lin, H. Mannila, G. Renganathan, and P. Smyth. Rule discovery from time series. In *Proceedings of the 1997 ACM SIGKDD International Conference*, ACM SIGKDD, 1997.
8. D.Brown and R.Jennings. On technical analysis. *Review of Finance Studies*, 2:527–551, 1989.
9. Federico Michele Facca and Pier Luca Lanzi. Mining interesting knowledge from weblogs: A survey. Technical Report 2003.15, Dipartimento di Elettronica e Informazione. Politecnico di Milano., April 2003.
10. J. Farrell. *Portfolio Management: Theory and Application.* McGraw-Hill, 1997.
11. P. Fraternali, M. Matera, and A. Maurino. Conceptual-level log analysis for the evaluation of web application quality. In *Proceedings of LA-Web'03, Santiago, Chile, November 2003.* IEEE Computer Society, 2003.
12. T.-C. Fu, F.L. Chung, V. Ng, and R. Luk. Pattern discovery from stock time series using self-organizing maps. In *Proceedings of the 1997 ACM SIGKDD International Conference*, ACM SIGKDD, 2001.
13. G.Ramazan. The predictability of security returns with simple trading rules. *The Journal of Empirical Finance*, 5:347–359, 1998.
14. T. Imielinski and H. Mannila. A database perspective on knowledge discovery. *Coomunications of the ACM*, 39(11):58–64, November 1996.
15. A. Ito. Empirical evaluation of technical analysis: A synthesis. Technical report, International University of Japan, November 1999.
16. M.C. Jensen. Random walks and technical theories: Some additional evidence. *The Journal of Finance*, (25):469–482, 1970.
17. R. Kohavi and R. Parekh. Ten supplementary analyses to improve e-commerce web sites. In *Proceedings of the Fifth WEBKDD Workshop: Webmining as a premise to effective and intelligent Web Applications*, ACM SIGKDD, Washington, DC, USA, 2003. Springer-Verlag.
18. L.Blume, D.Easley, and M.O'Hara. Market statistics and technical analysis: the role of trading volumes. *The Journal of Finance*, 49:153–181, 1994.
19. A. W. Lo, H. Mamaysky, and J. Wang. Foundations of technical analysis: Computational algorithms, statistical inference, and empirical implementation. *The Journal of Finance*, LV(4):1705–1765, August 2000.
20. R. Meo, G. Psaila, and S. Ceri. An extension to SQL for mining association rules. *Journal of Data Mining and Knowledge Discovery*, 2(2), 1998.
21. Nicolas Pasquier, Yves Bastide, Rafik Taouil, and Lotfi Lakhal. Efficient mining of association rules using closed itemset lattices. *Inf. Syst.*, 24(1):25–46, 1999.
22. Nicolas Pasquier, Yves Bastide, Rafik Taouil, and Lotfi Lakhal. Mining bases for association rules using closed sets. In *Proceedings of the 16th International Conference on Extending Databases*, IEEE, 2000.
23. P. Pirolli, J. Pitkow, and R. Rao. Silk from a sow's ear: Extracting usable structures form the web. In *Proc. of CHI 96 Conference.* ACM Press, April 1996.
24. M. Pring. *An introduction to Technical Analysis.* McGraw-Hill, 1997.

25. John R. Punin, Mukkai S. Krishnamoorthy, and Mohammed J. Zaki. Logml: Log markup language for web usage mining. In R. Kohavi, B. Masand, M. Spiliopoulou, and J. Srivastava, editors, *WEBKDD 2001 - Mining Web Log Data Across All Customers Touch Points, Third International Workshop, San Francisco, CA, USA, August 26, 2001. Revised Papers*, volume 2356 of *Lecture Notes in Computer Science*, pages 88–112. Springer, 2002.
26. Jaideep Srivastava, Robert Cooley, Mukund Deshpande, and Pang-Ning Tan. Web usage mining: Discovery and applications of usage patterns from web data. *SIGKDD Explorations*, 1(2):12–23, 2000.
27. M. Teltzrow and B. Berendt. Web-usage-based success metrics for multi-channel businesses. In *Proceedings of the Fifth WEBKDD Workshop: Webmining as a premise to effective and intelligent Web Applications*, ACM SIGKDD, Washington, DC, USA, 2003. Springer-Verlag.
28. R. Wille. Concept lattices and conceptual knowledge systems. *Computers and Mathematics with Applications*, 23:493, 1992.
29. Mohammed Zaki. Mining non-redundant association rules. *Data Mining and Knowledge Discovery*, 9:223–248, 2004.