Design of a data mining framework to mine generalized association rules in a web-based GIS

Sebastian Hähnel, Johannes Hauf, and Thomas Kudrass

***Abstract*— A Geographic Information System allows to create and manage spatial data. Having many public users who create and edit objects in geographic maps, the question of data quality arises. We introduce a data mining framework for generating association rules. During data maintenance users in different roles can be supported by showing hints and proposals with the help of those rules.**

I. Introduction

In the last years the number and the importance of online maps providing address lookup and routing services significantly increased. However, these services do not cover the increasing number of highly customized regional maps i.e. for tourism industry. This leads to numerous proprietary software solutions rendering interactive maps with objects that are linked to further information.

Business partners request applications with an user-friendly web-based interface to be able to administrate the objects in their maps on their own. Future developments may include to open the group of users sharing their spatial content to a public audience. At this point data quality will play an important role. Based on the data entry the system should be able to assist the user and to verify his submissions.

This work uses association rules to implement those mechanisms. At diverse moments during the runtime the application tries to find appropriate rules to provide the user with detailed information. Our data mining framework holds several independent modules that generate association rules for different purposes. The system periodically initiates the process and finds association rules based on the system's current data.

II. Preliminaries

A. Geographic Information Systems

Geographic Information Systems (GIS) are described by different definitions. [5] characterizes a GIS as a computer system for editing, storing, analyzing and displaying all the data that form a part of the earth's surface and its related data such as technical and administrative buildings as well as geoscientific, economic and environmental situations. GIS contains both spatial data and associated attributes.

In a regional area the map objects can be assigned to different topics. Most often objects of the same type (street, surface area) or of the same view (sanitation, eletricity network) belong to the same *layer*.

The GIS embedding the data mining framework uses *categories* to organize the *points of interest (pois)*. While the categories are different layers on the map, the pois represent the objects in the map. Users are able to edit the map directly. The categories may have a hierarchical structure like a taxonomy. That means that each category may have several sub-categories and only one super-category. Every category defines a set of attributes that are inherited by sub-categories. Because of that every poi has spatial data describing its location and attributes giving additional information. The GIS runs in an Apache Tomcat Servlet Container and thus is realized in Java.

The earth's surface can be described by geometric forms. A topographical map contains polygons and lines in different colors where objects with the same color belong to the same type. In our system you can import spatial geometries and assign them to a certain *topographic class*. I.e. you can import lines representing all major roads of a region and assign them to the topographic class Major Roads.

B. Spatial databases and the OpenGIS SFSQL

Spatial geometries can be stored in spatial databases such as DB2 Spatial, Oracle Spatial, PostgreSQL/PostGIS or even MySQL5. The *OpenGIS SFSQL (Simple Features Specification for SQL)* [9] describes how to extend relational databases respectively how to handle spatial data with SQL.

Spatial geometries such as points, lines, polygons or collections are mapped onto special geometry columns as part of database tables. Associated non-spatial attributes are stored within columns that are of standard SQL92 data types. A set of functions and operators allow to create, query, update, analyze and transform spatial features.

C. Generalized association rules

The mining of association rules as a data mining technique is a widely described method and was first introduced in [1]. Given a set of items $\mathcal{I} = \{i_1, i_2, ...i_n\}$, each combination $A \subseteq \mathcal{I}$ of items is called itemset. Let $\mathcal{D}$ be a transaction table where each transaction T is an itemset such that $T \subseteq \mathcal{I}$. The implication $A \rightarrow B$ with $A \subset \mathcal{I}$, $B \subset \mathcal{I}$ and $A \cap B = \emptyset$ is called *association rule* with the *antecedent* A and the *consequent* B. The *support* of an itemset A is described by the percentage of the transactions in $\mathcal{D}$ that contain A. The support of a rule $A \rightarrow B$ is defined by the percentage of the transactions in $\mathcal{D}$ that contain $A \cup B$. The *confidence* of

Sebastian Hähnel and Thomas Kudrass are with the Department of Sciences, Hochschule für Technik, Wirtschaft und Kultur (University of Applied Sciences), Leipzig, Germany (phone: +49-341-3076-6420; fax: +49-341-3076-6381; email: {shaehne, kudrass}@imn.htwk-leipzig.de).

Johannes Hauf, ars navigandi GmbH, Munich, Germany (phone: +49-89-8298-9165; fax: +49-89-8921-7646; hauf@arsnavigandi.de).

the rule is defined by the percentage of transactions in $\mathcal{D}$ containing A that also contain B.

The problem of mining association rules in a given transaction table $\mathcal{D}$ is finding all rules that have a greater support and confidence than the user-defined thresholds *minimum support* and *minimum confidence*. The problem is divided into two steps:

1) Finding all frequent itemsets in $\mathcal{D}$. That is: Find all itemsets that have a greater support than the user-defined minimum support treshold.
2) Using the frequent itemsets to generate all rules that satisfy the user-defined minimum confidence treshold.

The performance of the mining process mainly depends on the first step. There are several approaches to find all frequent itemsets. A systematization of established algorithms and a detailed comparison between them is given in [8]. Two of the most popular algorithms are Apriori [2] and FPgrowth [6].

Mining generalized association rules was first introduced in [3]. Another approach can be found in [7]. The problem of mining association rules is extended by including a taxonomy over the items. Let $\mathcal{T} = \mathcal{T}(V, E)$ be a directed acyclic graph (DAG) with the nodes V and the edges E. Let $\mathcal{T}$ be a taxonomy of the set of items $\mathcal{I}$ with $V = \mathcal{I}$ and $E \subseteq \mathcal{I} \times \mathcal{I}$. If there is an edge from $\hat{a}$ to a in the transitive-closure of $\mathcal{T}$, $\hat{a}$ is called the *ancestor* of a (and a the *descendant* of $\hat{a}$). Let A be an itemset with $A \subseteq \mathcal{I}$. Let $\hat{A}$ be an itemset that results from replacing one ore more items in A with their ancestors. Then $\hat{A}$ is called the *ancestor* of A (and A the *descendant* of $\hat{A}$).

A transaction T of the transaction table $\mathcal{D}$ *supports* the item $a \in \mathcal{I}$ if a or any of its ancestors $\hat{a}$ is in T. A transaction T supports the itemset A if it supports any of the items in A.

A *generalized association rule* or *multilevel association rule* is an association rule $A \rightarrow B$ that contains items from a taxonomy, and no item in B is an ancestor of any item in A. The definitions for support and confidence are extended to include the ancestors of A and B.

Beside support and confidence additional interestingness measures like *lift*, *conviction* and *rule interest* can be applied [8]. In [3] an interestingness measure for generalized association rules is introduced. Given a taxonomy over the items, the measure is based on ”expected” values for support and confidence of $A \rightarrow B$. Both are computed regarding the rule's closest parent rule $\hat{A} \rightarrow \hat{B}$ and considering the distribution of the items in the taxonomy. Similar to other interestingness measures a user-defined minimum treshold can be defined to prune useless rules.

III. Main Results

A. Design of the framework

The main intention of the framework is to manage several modules, where each module generates association rules for a specific purpose. The modules should be configurable and retrofitable during runtime and be able to be associated to each of the existing algorithms that generate the association rules. The mining process is initiated periodically and starts all mining modules sequentially.

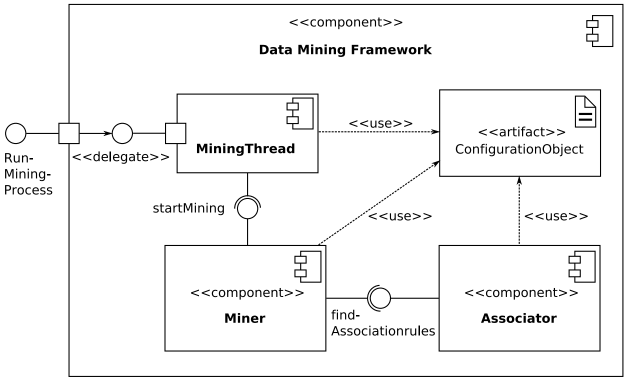

Fig. 1. Component view of the data mining framework

Each module consists of a *miner* and a linked *associator*. A configuration object holds the class names of the miners and associators. With the help of the Java Reflection API we can instantiate the miners and associators without restarting the virtual machine. To be able to work with these classes interfaces have to be designed. Figure 1 shows the component view of the data mining framework.

A miner's task is to create the *transaction table* and to start the associator that mines the rules. The transaction table is stored as a table within the database where each row consists of the transaction id, the item as an integer value and the number of the partition. Partition numbers later will be useful when the transaction table will be divided into several partitions. The items in the transaction table represent entries in database tables from the GIS. The section about the property vectors will explain how most of the items will be produced. Since the items are kept as numbers there must be a mapping between the database tuples and the integer values. A special data structure undertakes the task of doing the mapping.

Using the transaction table as input, the associator implements an algorithm to generate the association rules. The association rules are stored in another database table which is the *temporary rule table*. When the associator has finished mining the association rules the miner scans the temporary rule table and, after translating back the item values, stores the rules in the database permantly.

B. Property vectors for the pois

Most of the data mining modules will create rules that somehow are related to the objects in the maps. Therefore the transaction table mostly will contain items that are the properties of the pois, and each transaction may contain the properties of one poi. It is useful to create the property vectors for the pois before the mining process is started. A poi's property vector consists of:

- *A) Attribute vector*
- *B) Geographic property vector*

```
procedure gen_geovec()
 for each category c
    for each poi p $\in$ c.pois
      for each geometry_relationship_function grf
        for each topography_class tc {
          v := execute_function(grf, p, tc.spatial_tables);
          add_to_geopropvec(p.id, grf.id, tc.id, v);
        }
```

Fig. 2. The synthesis of the geographic property vector

A): All attribute data of a poi that is chosen for the data mining process builds the attribute vector. Remember that the poi's category defines a set of attributes. The attributes can be of different data types (i.e. literals, numbers, dates, lists ...). Numeric attributes must be discretized by defining intervals before they can be used in the mining process.

B): Based on the location and the shape of the poi and with the help of the spatial data that describe the topography of the region, a geographic property vector can be extracted. As mentioned above, the GIS uses different topography classes associated to spatial tables that describe the topography. A spatial database offers *geometry relationship functions* to analyze the relationship of geometries (i.e DISTANCE, INTERSECTS, TOUCHES, ...). The property vector of a poi results from using each of the geometry relationship functions with the pois spatial data and the spatial tables associated to each topography class. In case a geometric relationship function returns a numeric value, intervals have to be defined. The pseudo-code of this process in listed in figure 2.

C. How to handle taxonomies

The GIS hosting the data mining framework uses categories to organize the pois. In most cases the categories will have a hierarchical structure. Since the properties of the pois are subject of most of the rules, a taxonomy is available over the items.

The 'normal way' to include taxonomies would be to use an associator that can handle them. Therefore, an algorithm like in [3] or in [7] had to be implemented, and the interface between the miner and the associator had to be extended to hand over the taxonomy.

However, we employed another approach in our framework. The first step is to partition the transaction table. All transactions containing the same item of the taxonomy belong to the same partition. For each partition, the associator mines association rules ignoring the taxonomy. After that, each item in the taxonomy (that is: each category) is linked to a set of simple association rules. Next step is to generate the missing rules with the help of the taxonomy. Each rule consists of one itemset in its antecedent side and one itemset in its consequent side. By combining all itemsets of a rule, including their parent itemsets, missing rules can be created (by keeping them on their original side of the rule). A final scan of the transaction table is necessary to compute the interestingness measures like support and confidence. Furthermore, with the help of the expected values for support and confidence, redundant rules are pruned (see section II-C).

This approach has advantages over the 'normal way':

- As associator we can use a wrapper to include simple and established algorithms that are implemented in Java, like [4] and [10] or any other.
- Performance evaluations of algorithms that find frequent itemset show that the time elapsed per transaction is almost the same for a varying number of total transactions. Mining a single partition uses less memory than mining the whole transaction database at once. By splitting the transaction table into partitions and mining the partitions sequentially we can solve larger problems on the same machine (in nearly the same time).

D. Using the rules

As mentioned the association rules come in use at several different moments during the runtime of the GIS, depending on the intended purpose of the mining module that created the rules. The rules are bound to a specific module by an identifier so that only the desired rules are considered.

Using a rule mostly means determining if a certain object breaks the rule. To do that the following steps have to be processed:

1) Creating the property vector of the object the same way as the property vectors of similar objects were created during the data mining process.
2) Determining the application of each rule. A rule $A \rightarrow B$ applies if the object's property vector V contains all items that are in the rule's antecedent side, that means $A \subseteq V$.
3) Checking if each applying rule holds. A rule $A \rightarrow B$ is not broken if the object's property vector V contains all items that are in the rule's consequent side, that means $B \subseteq V$.

If a rule is violated a *rule break message* will be created. The message contains the broken rule and the items that caused the break of the rule.

E. Sample module and sample data

One sample module has been implemented. Its purpose is to answer the question: "How does the attribute vector of a poi affect its geographic property vector?" Based on the assumption that similar pois have a similar geographic property vector the user (editor) can identify misplaced pois. Every time a public user adds a poi or changes a poi's position or shape the system checks the new poi against existing rules and reports about rule violations. With the help of the rule break messages an editing user can easily identify outliers.

As sample region we downloaded the topography of the Bay Area, CA, USA, available at the Esri Data Downloader[1]. It is only a simple vector model of the area.

The sample data consists of about one hundred harbours, cinemas, theaters, bars and clubs. Figure 4 shows a rule that has been found for a special group of harbours. The values of the distance function have been discretized into few intervals.

[1] http://arcdata.esri.com/data_downloader/DataDownloader?part=10200

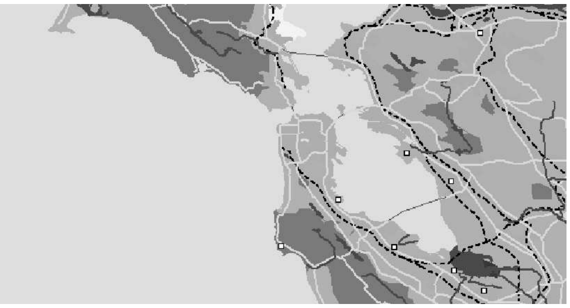

Fig. 3. Sample data: Bay Area, CA, USA

support:	0.233
confidence:	0.833
antecedent:	Categorie: Harbors Body of Water: Oakland inner Harbor
consequent:	Contains(land) = 1 Contains(Urban Area) = 1 Intersects(Land) = 1 Intersects(Urban Area) = 1 Distance(Greenland) = 3000 <= x and x < 30000 Distance(Major Railroad) = 3000 <= x and x < 30000 Distance(Coastline) = x < 3000 Distance(Airport) = 3000 <= x and x < 30000 Distance(Water Courses) = 3000 <= x and x < 30000 Distance(Trees) = 3000 <= x and x < 30000

Fig. 4. An association rule for the Oakland's inner harbours

The hierarchical structure of the categories generalizes the sub-categories CINEMA, THEATER and BARS AND CLUBS to NIGHTLIFE AND ENTERTAINMENT. Figure 5 shows a rule that has been found for all sub-categories of NIGHTLIFE AND ENTERTAINMENT. Note that the rule has been generalized to a higher level in the taxonomy.

IV. CONCLUSIONS

A. Evaluation

The association rules that have been mined only describe the data in the GIS rather than the real world. The intention of the rules is to support the user maintaining the data, whereas the user expects the rules to describe the reality. Hence the usefulness of the rules depends on how good the data describe the reality. The following factors affect the rules:

- the accuracy level of the topographic model
- the data preprocessing of the poi's attribute data (selection, discretization)
- the data preprocessing of the poi's geographic property vectors (selection, discretization)
- the way a miner synthesises the transaction table
- the tresholds for minimum support, confidence and interestingness for the generalized association rules

support:	0.543
confidence:	0.543
antecedent:	Category: Nightlife and Entertainment
consequent:	Contains(Land) = 1 Contains(Urban Area) = 1 Intersects(Land) = 1 Intersects(Urban Area) = 1 Distance(Greenland) = 3000 <= x and x < 30000 Distance(Coastline) = 3000 <= x and x < 30000

Fig. 5. An association rule for the nightlife in the Bay Area

While the accuracy level of the topographic model may be a financial problem, the other points need expertise. With the help of experience and data domain knowledge an expert is able to configure the data mining framework for satisfying results.

In some cases it is desired that users violate the rules to correct the data model.

B. Further developments

The data mining framework needs to be expanded. There are more data mining modules imaginable.

- A search function that associates keywords to map objects will be implemented in the GIS. A new module can establish criteria for the occurrence of the keywords. With the help of association rules the system can propose the keywords to the user editing a map object.
- An interactive evaluation system enabling the user to evaluate map objects with subjectiv criteria can guess the user's evalutaion of new objects with the help of association rules. It may find correlations between the object's properties and environment and the evaluation results. Such subjective criteria can be how romantic, exciting or suitable for a target group a hotel or any other map object is. This may be of great interest for tourism industry.

A performance improvement can be achieved by hosting the data mining framework and the GIS on different machines and mirroring the database to both servers.

ACKNOWLEDGMENT

This paper summarizes Sebastian Hähnel's thesis that was mainly carried out at the ars navigandi GmbH, Munich, Germany, supported by the Software-Offensive Bayern.

REFERENCES

[1] AGRAWAL, R. ; IMIELINSKY, T. ; SWAMI, A.: Mining association rules between sets of items in large databases. In: *Proceedings of the ACM SIGMOD Conference on Management of data*. Washington, D.C.: May 1993, pp. 207-216.

[2] AGRAWAL, R. ; SRIKANT, R.: Fast Algorithms for Mining Association Rules. In: *Proceedings of the 20th International Conference on Very Large Databases (VLDB 94)*. Santiago, Chile: June 1994, pp. 487-499.

[3] AGRAWAL, R. ; SRIKANT, R.: Mining Generalized Association Rules, In: *Proceedings of the 21st VLDB Conference*. Zurich, Switzerland, 1995, pp. 407-419.

[4] CRISTOFOR, L.: *ARtool Project*, http://www.cs.umb.edu/˜laur/ARtool/: reviewed May 2005.

[5] BARTELME, N.: *Geoinformatik: Modelle, Strukturen, Funktionen*. Springer, Berlin, 2000.

[6] HAN, J. ; PEI, J. ; YIN, Y.: Mining Frequent Patterns without Candidate Generation. In: *Proceedings of the 2000 ACM-SIGMOD International Conference on Management of Data*. Dallas, USA: 2000, pp. 1-12 .

[7] HAN, JIAWEI ; FU, YONGJIAN: Discovery of Multiple-Level Association Rules from Large Databases. In: *Proceedings of the 21st VLDB Conference*. Zurich, Switzerland: 1995, pp. 420-431.

[8] HIPP, J.: *Wissensentdeckung in Datenbanken mit Assoziationsregeln*. Dissertation, Eberhard-Karls-Universität Tübingen, 2003.

[9] OPEN GEOSPATIAL CONSORTIUM, INC.: *OpenGIS Simple Features Specification For SQL*. Revision 1.1, OpenGIS Project Document 99-049. 1999.

[10] WITTEN, I. H. ; FRANK E.: *Data Mining Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers, 2005.