NETLAKE toolbox for the analysis of high-frequency data from lakes

Factsheet #7

Knowledge Discovery in Databases - Data Mining

Ivanka Pižeta

Objective

Knowledge discovery in databases (KDD) (Fayyad et al. 1996) is the process of identifying valid, novel, potentially useful, and ultimately understandable patterns or models in data. Data mining (DM) is a step in the knowledge discovery process consisting of particular data mining algorithms that find patterns or models in data. Techniques involved in data mining represent a blend of statistics, pattern recognition and machine learning. An online application is presented where datasets are uploaded to apply KDD.

Specific application

As the objective of knowledge discovery is very broad, the outcome of KDD depends on the expert knowledge of the user who collects the data. The user must define and understand the problem, understand, prepare and model the data, evaluate the results and implement them (an example in this factsheet will try to help in understanding the principles of this work). KDD could give suggestions for further collection of data, for modification of an experiment, and for more specific statistical analyses.

Background

The background of the method is algorithms based on Boolean algebra (see e.g.: http://www.ee.surrey.ac.uk/Projects/Labview/boolalgebra) representing the Inductive Learning by Logic Minimization (ILLM) system. Basic understanding of data mining principles is required (see introductory chapters in Fayyad et al. 1996).

Type of data and requirements

Data should be organized in a tabular form. Each row is an example/case, and each column is an attribute/parameter. The first row contains names of the attributes. The maximal number of examples in the basic program is 250 with up to 50 attributes (except in http://dms1.irb.hr

where more examples are allowed). They could **be mixed, numerical and categorical data**, and it is not necessary that each case (row) has an attribute value (column). The ability to work with missing data is an added advantage of KDD.

Basic procedures

In order to approach the terminology of DM, the basic procedure will be explained through the example about smokers, given below. A parallel example would be the occurrence of lake stratification or algal blooms.

The procedure starts by defining the problem: if someone is interested in the problem of smokers (or the occurrence of lake stratification or algal blooms) and wants to find out their main characteristics and how they are different from non-smokers (or no occurrence of lake stratification or algal blooms), one has to collect data (attributes) of every case (person/lake) in the population of interest, which includes both smokers and non-smokers, their age, sex, education, profession, income and so on (lake characteristics like surface area of the lake, depth, temperature, wind, nutrients, oxygen...). The result of the data collection phase is a data table in which every object (person/lake) corresponds to one row of a table, described by a set of attributes (columns). For unknown attribute values '?' is used instead. In the 'smoker' problem the attribute containing the information if a person is a smoker or non-smoker (in the lake problem if a lake is stratified/non-stratified or an algal bloom is present/absent) presents the **target attribute** (with values “YES” or “NO”). It means that we are interested in models in which we relate the property “smoker”/”stratification or algal bloom” to other attributes of the person/lake. Every data mining task must have one, and only **one target attribute at a time**. All other attributes are **input attributes**, which are used to build the model of the smoker/a stratified lake/algal bloom present.

After we have selected the target attribute, we must select also the **target class**. In our domain, the target attribute has two classes: smokers and non-smokers. We can select any of these classes as the **target (positive) class**. The other class is the non-target or **negative class**. The result of the data mining process is one or more models (rules), which describe some of the most important subgroups of the target (positive) class. Models describe differences between the target and the non-target (negative) class. Input attributes are used in model descriptions. It must be noted that **existence of examples in both target and non-target classes is mandatory** because the object of induction is the search for differences between the classes.

The table with the collected data for N cases will have N+1 rows (the first row contains attribute names) and M+1 columns (M input attributes and one target attribute). The target attribute is marked by an exclamation mark (!) in front of the first character of its name in the first row, and the target positive class is marked by an exclamation mark (!) in front of each positive class value (see Table 1). Exclusion of an attribute from the calculation is accomplished by putting a question mark (?) in front of its name in the first row. In another session, another attribute could be assigned to be a target one. Also, another set of attributes could be included or excluded from the calculation (in lake science, a target attribute could be either occurrence

of stratification or occurrence of an algal bloom). As discussed below, the entire process of so-called model induction depends on the quality and quantity of data put into the table.

NAME	AGE	SEX	EDUCATION	PROFESSION	WEIGHT	INCOME	?SMOKER	!class_SMOKER
Jan	30	M	LOW	worker	27.3	14000	!YES	!1
John	55.5	M	MIDDLE	worker	90	20000	NO	0
Clara	?	F	HIGH	teacher	65.2	1000	NO	0
Mary	18	F	MIDDLE	student	55.1	0	NO	0
Tom	70	M	HIGH	?	60	9000	!YES	!1
Bill	35	M	MIDDLE	prof	33	16000	NO	0
Steve	42.2	M	LOW	driver	27	7500	!YES	!1
Marc	29	M	?	waiter	31	8300	!YES	!1

Table 1. A simple example of a table prepared for data mining ('smokers' example).

Data can be prepared in Excel, and then saved as a *.txt* file as requested by the software. The data is then uploaded by the program to the server where the calculation will be done and results in the form of models (rules) will be displayed.

It should definitely be noted that the rule obtained, accurately characterises the difference between the examples describing smokers and non-smokers (Table 2). All smokers are men and have an income below 15000. In other words, all non-smokers are either women, or men who earn more than 15000. From the standpoint of quality in the learning set and interpretability of results we can be quite satisfied with the result. We cannot, however, be satisfied with the overall result, as we know that actually there are a large number of women who smoke. For the same reason we can also expect that the predictive quality of the resultant model will be poor.

Induction results:
The result is the following model for the positive class of the target attribute class_SMOKER

SUBGROUP A
true positive rate (sensitivity*) 100.0%
true negative rate (specificity**) 100.0%

SUBGROUP A has 2 conditions which both must be satisfied:
attribute SEX is equal M
attribute INCOME is less than 15000.00

Table 2. Resulting model for the positive class of the target attribute class_SMOKER.

Note that the model output includes information on its sensitivity and specificity. **Sensitivity** is a relative number representing the number of correctly predicted positive examples in respect to the total number of positive examples in the input data file. High sensitivity is a greatly appreciated property of every good model, especially if high sensitivity can be obtained together with high specificity. **Specificity** is a relative number representing the number of correctly predicted negative examples in respect to the total number of negative examples in the input data file. High specificity is a necessary property of reliable data models. Many

applications require specificity equal or very near to 100%. Only in situations which require general models or models with high sensitivity, can specificity below 80% be tolerated.

By increasing the **generalization level** (a parameter selectable during data upload) the user can try to induce models with better sensitivity, but typically, models induced in this way will have worse specificity. It is always worth trying this possibility because the decrease of the specificity may be less significant than the gain obtained in sensitivity. By decreasing the **generalization parameter** (a parameter selectable during data upload) the user can try to induce models with better specificity, but typically, models induced in this way will have lower sensitivity.

The poor quality of this example model is the result of problems in data collection. In our set of eight collected examples there is not a single woman smoker. Methods of machine learning as a source of information about the world have only the data we give them in the form of examples. In this case there was no theoretical chance to construct a proper model of women smokers. The only solution is to expand the set of examples and repeat the procedure of induction models.

The conclusion is that the quality of the induced model depends entirely on the quality of the input data. This equally applies to the choice and the amount of available examples as well as the selection and quantity of the attributes which describe examples.

Pitfalls and tips

As it is time consuming to add exclamation marks in front of each target attribute value assigned a positive class, it is more convenient to form additional double columns for each attribute that will be assigned a target one (and keep them inactive by "?" in front of their name). In this column, "!1" is put for a positive class, and "0" for negative classes. When this column is assigned to be a target attribute (by "!" in front of its name), the true one is excluded by "?" (Think/try what would happen if not!). If, for example, we want to assign a positive class to an attribute having values smaller than a certain number, then we first apply SORT BY that attribute function in Excel, then create this double column of "!1!" and "0", along with our decision of what a positive class is. In another session (run), it can be easily changed, either by choosing a new target attribute or by shifting the border of positive class in the same target attribute by rearranging "!" and "?" signs.

Further reading

Key References:
An introduction can be found on the same webpage as the program itself:
http://dms.irb.hr/index.php and http://dms.irb.hr/tutorial/tut_applic_ref.php

For the fundamentals of data mining see:
Fayyad U., Piatetsky-Shapiro, G., Uthurusammy, R. (Eds.) 1996. Advances in Knowledge Discovery and Data Mining. AAAI Press/ The MIT Press. Massachusetts.

Other useful references:

Pyle, D. 1999. Data Preparation for Data Mining. Morgan Kaufmann. San Francisco.

Hofsheimer, M., Siebes, A.P. 1994. Data Mining: The Search for Knowledge in Databases. Technical Report. Centre for Mathematics and Computer Science. Amsterdam.

Witten, J., Eibe, F. 2000. Data Mining: Practical Machine Learning tools and techniques wit Java implementations. Morgan Kaufmann. San Francisco.

Weiss, S., Indurkkhya, N. 1998. Predictive Data Mining - A practical guide. Morgan Kaufmann. San Francisco.

Berry J.A., Linoff, G.S. 2000. Mastering Data Mining: the art and science of customer relationship management. John Wiley & Sons. New York.

Fürnkranz, J., Gamberger, D., Lavrač, N. 2012. Foundations of Rule Learning. Springer. Heidelberg.

Code

On-line data treatment is available. A tutorial and detailed instructions on how to prepare the data in a table, how to upload it and get the results is available at: http://dms.irb.hr.

Contact details

Ivanka Pižeta. Ruđer Bošković Institute, Zagreb, Croatia.
pizeta@irb.hr

Also, find contact details of the authors of the program on the web page.

Suggested citation

Pižeta, I. 2016. Knowledge Discovery in Databases - Data Mining. In Obrador, B., Jones, I.D. and Jennings, E. (Eds.) *NETLAKE toolbox for the analysis of high-frequency data from lakes* (Factsheet 7). Technical report. NETLAKE COST Action ES1201. pp. 35-39. http://eprints.dkit.ie/id/eprint/538.

Acknowledgement

This factsheet is based upon work from the NETLAKE COST Action ES1201, supported by COST (European Cooperation in Science and Technology).