DATA MINING CUP Competition 2013

Prediction of orders

The topic of this year's DATA MINING CUP (in short: DMC) Competition is the prediction of orders in an online shop. In the practice such a prediction provides some advantages. So, for example, in case of a significantly high predicted probability for an order it is possible to recommend top-selling products to the visitor in order to achieve upselling. In case of an accordingly lower probability coupons could be offered to the visitor in order to raise the motivation for a purchase. For an appropriate calculation of the ordering probabilities, methods of data mining can be applied. There is a multitude of such methods, and they should be able to answer the question about a possible order to a satisfying degree.

Scenario

The visit of an online shop by a possible customer is also called a session. During a session the visitor clicks on products in order to see the corresponding detail page. Furthermore, he possibly will add or remove products to/from his shopping basket. At the end of a session it is possible that one or several products from the shopping basket will be ordered. The activities of the user are also called transactions. The goal of this year's DMC contest is to predict whether the visitor will place an order or not on the basis of the transaction data collected during the session.

Tasks

This year's DMC Competition comprises two tasks.

In the first task historical shop data are given consisting of the session activities inclusive of the associated information whether an order was placed or not. These data can be used in order to subsequently make order forecasts for other session activities in the same shop. Of course, the real outcome of the sessions for this set is not known. Thus, the first task can be understood as a classical data mining problem.

As in the previous years the second task deals with the online scenario. In this context the participants are to implement an agent learning on the basis of transactions. That means that the agent successively receives the individual transactions and has to make a forecast for each of them with respect to the outcome of the shopping cart transaction. This task maps the practice scenario in the best possible way in the case that a transaction-based forecast is required and a corresponding algorithm should learn in an adaptive manner.

Data

For the individual tasks anonymised real shop data are provided in the form of structured text files consisting of individual data sets. The data sets represent in each case transactions in the shop and may contain redundant information. For the data, in particular the following applies:

1. Each data set is in an individual line that is closed by “LF” (“line feed”, 0xA), “CR” (“carriage return”, 0xD), or “CR” and “LF” (“carriage return” and “line feed”, 0xD and 0xA).
2. The first line is structured analog to the data sets but contains the names of the respective columns (data arrays).
3. The header and each data set contain several arrays separated by the symbol “|”.
4. There is no escape character, and no quota system is used.
5. ASCII is used as character set.
6. There may be missing values. These are marked by the symbol “?”.

In concrete terms, only the array names of the attached document “*features.pdf*” in their respective sequence will be used as column headings. The corresponding value ranges are listed there, too.

The training file for task 1 of the DMC Competition (“*transact_train.txt*“) contains all data arrays of the document, whereas the corresponding classification file (“*transact_class.txt*”) of course does not contain the target attribute “order”.

In task 2 data in the form of a string array are transferred to the implementations of the participants by means of a method. The individual fields of the array contain the same data arrays that are listed in “*features.pdf*” – also without the target attribute “order” – and exactly in the sequence used there.

Submission

The participants can submit their results up until 15 May 2013. The following task descriptions explain how to submit the results.

Task 1

Approach

For the first task historical data of about 50,000 sessions are known by means of which a model for the prediction of orders can be learned (offline learning). The target attribute "order" of the sessions is known here, and it will be indicated by the parameter value "y" for a placed order and by the parameter value "n" for no order placed. For about 5,000 sessions, each time it is to be evaluated whether an order is placed or not. For this purpose for each session a forecast is to be made the value of which is within the closed interval [0,1]. Here, 0 indicates a non-purchase, and 1 a purchase. The error with respect to the real outcome of the session should be as small as possible.

Submission

In order to submit the data of the solution a file with the following format is to be used:

Column name	Description	Range of values
sessionNo	Running number of the session	Natural number
prediction	Prediction whether an order will be placed	Floating point number from [0,1]

Each session number of the classification data must appear here exactly one time. Furthermore, the file should comply with the specifications defined in the paragraph "Data" insofar as they are applicable.

The result file must finally be submitted as e-mail attachment in form of a zip text file to dmc_task1@prudsys.de. The name of the zip file and of the included text file must be composed of the team name, the task name and the file type (zip or text):

"<Teamname>_task1.zip", (e.g. *TU_Chemnitz_1_task1.zip*),

and

"<Teamname>_task1.txt", (e.g. *TU_Chemnitz_1_task1.txt*).

The team name was submitted to the team leader together with the registration confirmation.

Evaluation

The submitted solutions will be evaluated and compared by means of the following error functional that is to be minimized:

$$E = \sum_i \left| order_i - prediction_i \right| .$$

Here, $order_i$ is the real outcome of session i (0 means no order, 1 means order completed), and $prediction_i$ is the predicted outcome of session i. Those teams whose error functionals reach the smallest values will win. In case of a tie the decision will be made by drawing lots.

Task 2

Approach

In the second task a Java program (agent) is to be implemented that will learn by means of transaction-based data. The agent should be able to predict the probability for an order in the course of a session after each individual transaction (online learning). Altogether about 30,000 sessions (corresponding to about 280,000 transactions) have to be predicted. The evaluation and the error functional are identical to those of task 1, but an evaluation is made for each transaction. For a valid solution the agent must implement the interface Reviewer.

Frame conditions for the implementation

The implementation must be made in the form of a Java class. The condition is that the latter can run on a PC with 1024 MB of assured heap space and a 1.6 GHz Intel Core Duo processor. The sandbox principle applies, i.e. the application must not establish a connection to external resources (e.g. http connections). Java libraries may be used, but they must be subject to an unrestricted license. The application must not create files, make outputs or send signals.

The interface contains the method *submitTransaction* that is called up after each transaction and transfers the data set of the transaction. This data set is realized as a string array – its individual components and the respective meaning can be found in the file "*features.pdf*" as described above in the paragraph "*Data*". The method *submitTransaction* must return the evaluation of the session as Double value (from the interval [0,1]).

Additionally the interface contains the method *submitOutcome* that is to be implemented, too, and that will be called up after the last transaction of a session. The passed parameter is a string with the value "y" if the last session led to a purchase, and with the value "n" if not.

The data from task 1 can be used to construct the solution of task 2.

```
public interface Reviewer {

      public double submitTransaction(String[] transaction);

      public void submitOutcome(String result);

}
```

Submission

The result file must be sent as one zip file to dmc_task2@prudsys.de. The maximum size of the zip file is 5 MB. The file name must be composed of the team name that is identical with the team ID, the task number *task2* and the file type (.zip):

"<Teamname>_task2.zip", (e.g. *TU_Chemnitz_1_task2.zip*).

The team name was submitted to the team leader together with the registration confirmation.

It is mandatory to indicate in the e-mail the complete name of the classes implementing the interface. If external Java libraries are used they have to be submitted, too. The implementation can be submitted as source code or as completely compiled bytecode (jar file). If the source code is submitted it must be possible to compile it with a current jdk1.7 version, and an Ant build file must be included. In the case of bytecode the latter must be executable in a current jre7 version. It would be helpful to add a short text file containing details as to the logic of the implemented algorithm.

Evaluation

A program of the prudsys DMC team successively calls up the individual agents with all associated transactions according to the history, and on the basis of the return value of the function *submitTransaction* incrementally calculates the error functional. That means that for each transaction the predicted probability is compared with the real outcome of the session and accordingly added up. If the sum of the response times of an agent exceeds one hour the solution is declared void and excluded from evaluation. The team whose error functional reaches the smallest value will win. In case of a tie the decision will be made by drawing lots.