

The Orange Customer Analysis Platform

Raphaël Féraud, Marc Boullé, Fabrice Clérot, Françoise Fessant, Vincent Lemaire

FIRSTNAME.LASTNAME@ORANGE-FTGROUP.COM

Orange Labs, 2 avenue Pierre Marzin, 22300 Lannion, France

Editor: Gideon Dror

Abstract

In itself, the continuous exponential increase of the data-warehouses size does not necessarily lead to a richer and finer-grained information since the processing capabilities do not increase at the same rate. Current state-of-the-art technologies require the user to strike a delicate balance between the processing cost and the information quality. We describe an industrial approach which leverages recent advances in treatment automatization and relevant data/instance selection and indexing so as to dramatically improve our capability to turn huge volumes of raw data into useful information.

Keywords: data mining, large scale learning, variable selection, instance selection

1. Introduction

According to (Fayyad et al., 1996), Data Mining is a nontrivial process which has to identify unknown, valid and potentially exploitable structures in databases. Several industrial speakers proposed a formalization of this process, under the form of a methodological guide named CRISP-DM for Cross Industry Standard Process for Data Mining Chapman et al. (2000). The CRISP-DM model suggest cutting any Data Mining process in six phases:

1. Business Understanding: This initial phase focuses on understanding the project objectives and requirements from a business perspective, and then converting this knowledge into a data mining problem definition, and a preliminary plan designed to achieve the objectives.
2. Data Understanding: The data understanding phase starts with an initial data collection and proceeds to an explanatory analysis to get familiar with the data and to identify data quality problems.
3. Data Preparation: This phase covers all activities to construct the final dataset, that will be fed into the modeling tool, from the initial raw or relational data (Pyle, 1999; Chapman et al., 2000).
4. Modeling: In this phase, various modeling techniques are selected and applied, and their parameters are calibrated to optimal values according to a success criterion.
5. Evaluation: Before proceeding to the final deployment of the model, it is important to more thoroughly evaluate the model, and review the steps executed to construct the model, to be certain it properly achieves the business objectives. At the end of this phase, a decision on the use of the data mining results should be reached.

6. Deployment: The deployment phase can be as simple as generating a report or as complex as implementing a repeatable data mining process. The model will be applied to a larger database (the target population) than the training database. In this phase it is important for the customer to understand up front what actions will need to be carried out in order to actually make use of the created models.

The CRISP-DM model is mainly a process guide for data mining project. Most of data miners agree to evaluate the time spend into the data preparation and deployment phases to be around 80% of the time spend in the entire project. The use of the statistical methods requires to represent the data under row (instances) and columns (feature, explanatory variables): a table. Now, to optimize the storage, the data are stored in relational databases, and this whatever is the studied phenomenon: genes, IP sessions, information on the customers...

During the data preparation phase, the first task of the analyst thus is to extract a table of the information system. This stage is not harmless because the number of potential representations of the relational data in table is gigantic. In practice, the analyst has to make an a priori for all the variables on which subsequent studies will be carried out. The consequence is that the loss of information due to the 'flattening' of the relational data is very important.

During the deployment phase, the elaborated model (beforehand) must be applied to all the target population, to produce a score for every instance. All the explanatory variables for all the instances must be built before the modeling phase. This phase is potentially very expensive when the number of instances and explanatory variables are important. The main commercial products of Data Mining propose platforms allowing to build and to deploy predictive models. Nevertheless, they do not offer satisfactory solution to exploit all the potential of the information contained in the initial relational database. In the industrial applications built on these software platforms, the number of explanatory variables used remains limited to some hundreds. Now the potential is simply of another order : with a simple relational model composed of a customer table and an usage table, the study of the number of uses by service types, per month, and per day of the week could lead to build up to 10000 explanatory variables!

In this paper we propose an automatization of the 'Data Preparation' phase driven by a powerful algorithm of variable selection (Boullé, 2007). In the next section the main elements of an architecture of data mining¹ and the algorithm to select the data representation are detailed. To make the models deployment easier, a method to extract, from a large database, a 'paragon' database² is presented. This paragon table is constituted of representative instances of explanatory variables useful to build the output of the model. This table is linked to the initial database with an index automatically elaborated. The deployment is then only a simple joint on all the instances. The algorithms allowing the extractions of the paragons and the index table are described Section 3. Then the Section gives 4 experimental results obtained on data from the French telecom company Orange. Finally the last section gives a conclusion.

1. French patent N 06 07965, date 09/01/2006

2. French patent N 05 05412, date 05/27/2005

2. Automation of Data Preparation and Modeling

2.1 Introduction

The purpose of variable selection is three-fold: to improve the classifier accuracy, to reduce the training and deployment time, and to ease the interpretability of the classifier (Guyon and Elisseeff, 2003). New techniques have been proposed to address the challenge of variable selection and modeling in high dimensions (Guyon et al., 2006), even when the number of variable is large compared to the number of instances. In order to benefit from these recent advances in an industrial context, we propose a platform dedicated to the automation of the data preparation, modeling and deployment phases of the data mining process. In this section, we describe the architecture of the platform and the languages exploited to easily extract representations with thousands of variables. We finally summarize the MODL method (Boullé, 2007), both reliable and time efficient, used to select the best representation. This method is efficiently implemented into the Khiops scoring tool (see www.khiops.com) which is used in the Customer Analysis Platform (CAP).

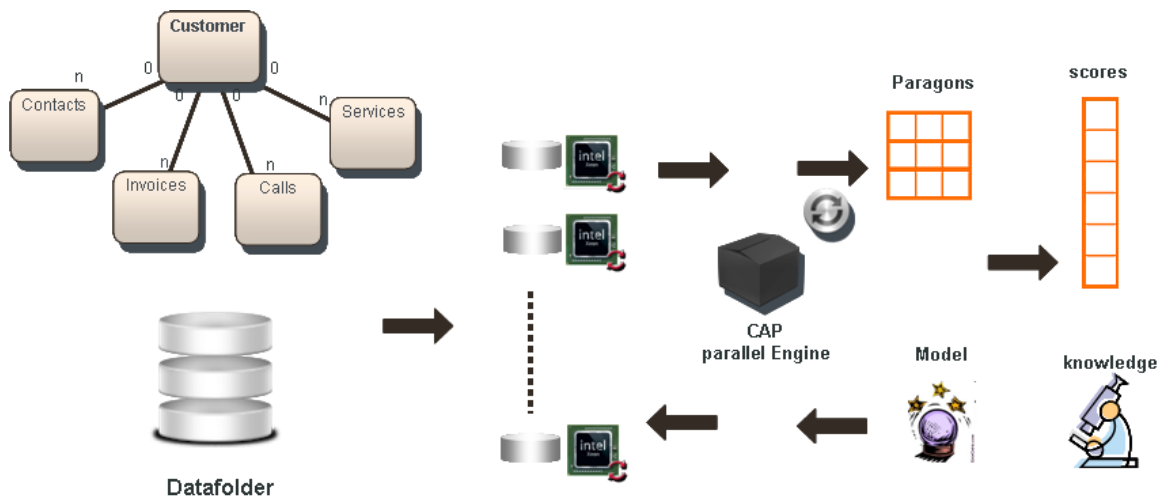


Figure 1: Principle : data are normalized and hashed in a star schema database. Using extraction languages, learning algorithms drive data preparation, modeling and instance selection. A server executes in parallel most of the process.

2.2 Platform Architecture

Unlike the current practice of data mining architecture, the explanatory variables are not designed and computed in a datamart. In our platform architecture, the input data from information system are structured, and stored in a simple relational database : the data folder (Figure 1). The explanatory variables are constructed and selected automatically for each specific marketing project. In order to be computed in parallel and in memory, the datafolder is hashed in small datafolders of size 1 Go.

The data folder model provides a unique view of the available input data sources, normalized according a star schema:

- The primary table is related to the marketing domain. For customer data analysis, this table contains all the fields directly connected to the customer, such as his name or address.
- The secondary tables have a 1-N relationship with the primary table. Each instance of the primary table may be related to a variable number of instances of a secondary table. For telecommunication data for example, the secondary tables contains the list of services, of usages of theses services, the call details.

This type of data modeling has a large expressiveness, suitable for many data mining projects. It offers an efficient trade-off between single-table data mining and full multi-relational data mining. The star schema allows to efficiently build many constructed variables, when the join key belongs to the primary key, whereas in a traditional data-warehouse, the construction of one single variable may involve multiple table joins. Finally, this star schema modeling allows the design of formatted data extraction languages, with the purpose of automation of the data mining process.

2.3 Data Extraction

The data extraction functionality of the platform is parametrized using three languages:

- a selection language to filter the instances,
- a construction language to build a flat instance x variables representation from the data folder,
- a preparation language to specify the recoding of the explanatory variables.

These languages are both simple enough to be automatically exploited by the process of variable selection and expressive enough to build a large variety of explanatory variables. Each language expression deals with at most two tables: the primary table plus eventually one secondary table. The join key always belongs to the primary table, and the selection and construction operands exploit the fields of any table, primary or secondary. For example, to build the number of usages of each service per weekday for all customers, one single language expression needs to be specified, with the use of the “Count” operator on the secondary table “Usage” with two operands “WeekDay(Date)” and “Label(ServiceId)”. It is then possible to specify up to thousands of variables to construct, using one single expression of the construction language.

2.4 Variable Selection

The platform architecture allows to easily build flat data tables with up to tens of thousands of constructed variables. In order to select the best representation, that is the best subset of informative variables, a powerful variable selection method is required, both robust and efficient. Two main approaches, filter and wrapper (Kohavi and John, 1997), have been studied in the literature. Filter methods consider the correlation between the input variables

and the output variable as a pre-processing step, independently of the chosen classifier. Wrapper methods search the best subset of variables for a given classification technique, used as a black box. Wrapper methods, which are time consuming (Féraud and Clérot, 2001; Lemaire and Féraud, 2006), are restricted to the modeling phase of data mining, as a post-optimization of a classifier. Filter methods are better suited to the data preparation phase, since they are time efficient and can be combined with any data modeling approach.

The most commonly used filter methods are based on statistical tests (Saporta, 1990) that consider the correlation between an input variable and the output variable, such as the chi-square test for categorical input variables, or Student or Fisher-Snedecor tests for numerical input variables. These statistical tests are easy to apply, but they suffer from serious limitations. They are restricted to a strong dichotomy between dependent and independent variables, which does not provide a reliable ranking of the input variables. They are also subject to strong constraints (minimum expected frequency in each cell of the contingency table for categorical variable, Gaussian distribution for numerical variables). Many alternative measures of associations between two variables have been studied in the context of decision trees (Kass, 1980; Breiman et al., 1984; Quinlan, 1993). These criteria are based on a partition of the domain of the input variable and the consideration of the dependence between the corresponding discretized input variable and the output variable. Supervised discretization methods split the numerical domain into a set of intervals and supervised grouping methods partition the input categories into groups. Fine grained partitions allow an accurate discrimination of the output classes, whereas coarse grained partitions tend to be more reliable. When the number of intervals of the discretization is a free parameter, the trade-off between information and robustness is an issue. In the MODL (Minimum Optimized Description Length) approach, supervised discretization (Boullé, 2006) (or grouping (Boullé, 2005)) is treated as a nonparametric model of conditional probability of the output variable given an input variable. The discretization is turned into a model selection problem and solved in a Bayesian way. First, a space of discretization models is defined. The parameters of a specific discretization are the number of intervals, the bounds of the intervals and the output frequencies in each interval. Then, a prior distribution is proposed on this model space. This prior exploits the hierarchy of the parameters: the number of intervals is first chosen, then the bounds of the intervals and finally the output frequencies. The choice is uniform at each stage of the hierarchy. Finally, the multinomial distributions of the output values in each interval are assumed to be independent from each other. A Bayesian approach is applied to select the best discretization model, which is found by maximizing the probability $p(\text{Model}|\text{Data})$ of the model given the data. Owing to the definition of the model space and its prior distribution, the Bayes formula is applicable to derive an exact analytical criterion to evaluate the posterior probability of a discretization model. Efficient search heuristics allow to find the most probable discretization given the data sample. Extensive comparative experiments report high performance. The case of categorical variables is treated with the same approach in (Boullé, 2005), using a family of conditional density estimators which partition the input values into groups of values.

The best discretizations and value groupings are optimized using the bottom-up greedy heuristic described in (Boullé, 2006). The algorithmic complexity of $O(n \log n)$ of this heuristic and the excellent reliability of this method allow to preprocess a very large number of variables, around 50000 in our experiments. A selective naïve Bayes classifier (Boullé,

2007), based on variable selection regularization and model averaging, is then exploited to discard redundant variables and fully automatically build effective scores.

3. Efficient Deployment

3.1 Principle

To produce scores, a model has to be applied for all instances on all explanatory variables. To speed up this process, a table of paragons containing representative individuals is extracted. The paragons are connected by an index to the whole population. Then, the scores of the whole population is obtained by a simple join between the table of the paragons and the index. This method of deployment is particularly effective when the model is deployed several times. For example for monthly marketing campaigns, only the reduced table of the paragons is built each month to produce the scores of all instances. This approach makes it possible to increase considerably the number of scores which can be produced on the same technical architecture.

3.2 Paragon Selection

The quality of the paragon table is crucial for the final performance of the system. A not representative paragon table leads to ineffective scores. A too large number of paragons increases computational cost. The table of paragons is drawn from the datafolder to be representative of relevant variables for the model. To produce and maintain online a sample of size n , Reservoir Sampling algorithm Vitter (1985)) can be used. An inclusion probability of $n/(t+1)$ is given for each tuple arrived at time t . An interesting property of this algorithm is that, when t tuples have been observed, all the t tuple have the same probability to be included in the reservoir: n/t . Biased versions of this algorithm exist, to take into account recent data (Aggrawal (2006)) or weighted data (Chaudhuri and Motwani (1999); Kolonko and Wasch (2004); Efraimidis and Spirakis (2004)). As the frequencies of explanatory variables are known from the variable selection stage, a biased version of Reservoir Sampling can be used to draw the paragons. However to capture multivariable dependences Weighted Reservoir Sampling needs to know the joint distribution of explanatory variables to set weights. The joint distribution can be approximated with frequencies when the number of explanatory variables is low. In our case, hundreds of variables are used by the model. To draw a small and representative set of instances, we use a local approximation of the Khi^2 criterion between the observed distribution of explanatory variables on the sample and on the whole population:

- the number of iterations is set to the number of paragons needed.
- At each iteration, the instance minimizing the Khi^2 in a sliding windows is chosen.

3.3 Data Indexing

The problem to be solved is simple to state: being given an individual, to find his nearest neighbor in the table of paragons. The $L1$ norm between the explanatory variables is used to evaluate the distance between instances. This task has to be done for all the instances of the datafolder. The search of nearest neighbors is an expensive operation.

Its naive implementation implies an exhaustive research among the paragons, therefore a complexity in $O(nmp)$, n being the number of instances, m the number of variables in space of representation and p the number of paragons. In order to accelerate the research of nearest neighbors, a compromise between speed and accuracy can be done : to find a paragon close to the nearest. It is precisely what the algorithm Locality Sensitive Hashing (Gionis et al., 1999) allows. It is based on a technique of hashing to select good candidates among the paragons to be close to the nearest. Then an exhaustive search is done on good candidates to find the paragon. Our implementation of this technique makes it possible to bring back the complexity of the search of nearest close to $O(nm\sqrt{p})$. It reduces the computational cost of a factor 300 per 100000 paragons, and leaves to the user the control of the compromise speed / performance.

4. Experiments

The scores built with our technology (CAP platform including the Khiops scoring tool) and built with the current model have been compared for several marketing campaigns of Orange Company in order to evaluate the reliability of our scores. The current model is built with KXEN (<http://www.kxen.com>) on a datamart containing 700 explanatory variables. We have collected data on about one million of customers between January and June 2005 to supply the platform. This information comes from decisional applications of Orange Company. The first four months have been used to build the customer profiles, the last two to compute the target variable. 20% of the customers are kept for the evaluation of the models. The performance of a model is measured thanks to the cumulative gain curve (Figure 2). It is a graphical representation of the advantage of using a predictive model to choose which customers to contact. The x-axis gives the proportion of the population to contact and therefore the cost of the campaign. The y-axis gives the percentage of the target population reached and therefore the gain of the marketing campaign.

The goal of the campaign presented below is to prevent a customer to cancel his ADSL subscription. The cumulative gain curves of several predictive models of cancellation are plotted Figure 2. The diagonal represents the performance of a random model. If we contact 20% of the population based on this random model, we will reach 20% of the customers who will cancel their subscription in next two months. When 20% of the population is contacted using the current model, 45% of the fragile customers are reached which represents a gain of 25% compared with a random targeting. The automation of the search of representation has led us to select a model based on 191 explanatory variables chosen among a set of 50000 variables. The model deployment is then achieved on all the instances with a variable number of paragons: 500, 5000, 15000 and also directly on the population. In the case of a direct deployment on all the instances, if we contact 20% of the population based on this targeting, 65% of the customers that will cancel their subscription in next two months are reached. Compared with the current technique, for the same number of mails, we are able to reach 20% of the target population more. This improvement remains true for the entire cumulative gain curve. When the technique of score deployment with paragons is applied, there is a loss of reliability which depends on the number of paragons used. The targeting comes close to the best when the number of paragons increases but it is also very costly. For example, when 5000 paragons are used to represent 1000000 customers, if 20% of the

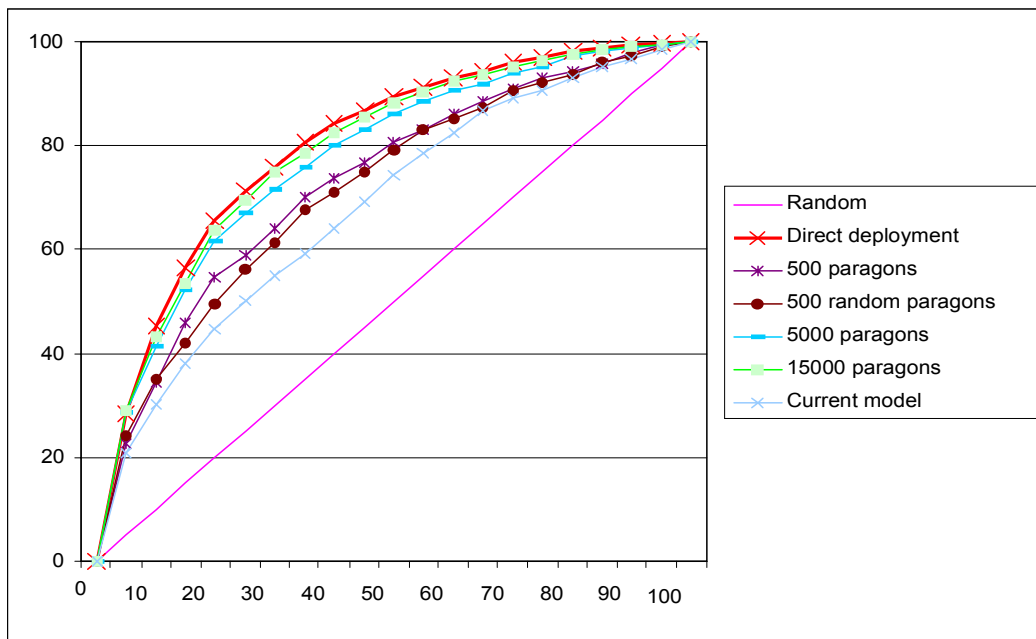


Figure 2: Lift curve of predictive models for ADSL cancellation.

population is contacted, 60% of the fragile customers are reached (+40% of gain compared with a random targeting and +15% compared with the current technique). With 15 000 paragons, the performances are similar to those of the direct deployment. To evaluate the quality of the algorithm of paragon selection, we have compared the performances obtained when the paragons are randomly selected and when the paragons are selected using a local optimization of the Khi^2 criterion. With 500 paragons ($K=200$, $M=100$), at the level of 20% of population, 50% of the target is reached for the random selection and 55% with the local optimization of the Khi^2 criterion (Figure 2).

The whole process of extraction of a paragon table from one million customers and a representation space of 50000 variables takes about 3 hours on a server with 16 processors and 32 Go of RAM. One third of the processing time is for the selection of the representation and two thirds is for the search and indexation of paragons. Once the paragons are available, the score production from the paragons table takes less than one minute. One processing hour is necessary in a direct deployment to generate a table of one million instances with 191 explanatory variables and apply the predictive model on this table. It is very efficient to use paragons for the deployment of a recurrent score like fragility scores or ADSL recruiting. For opportunist score as appetency to a specific offer, a direct deployment is better.

5. Conclusion

We have described a data-mining platform which allows to build predictive models using two orders of magnitude more explanatory variables than the current state-of-the-art, resulting in a dramatic improvement of performances. The platform relies on a novel archi-

ture which allows to leverage recent advances in treatment automatization and relevant data/instances selection and indexing. The processing time associated with data table flattening remains the main limitation to the exploration of an even larger data space. The efficient exploration of such huge spaces therefore requires the conception of an exploration technique guiding the flattening towards the most promising areas.

References

- C. Aggrawal. On biased reservoir sampling in the presence of stream evolution. In *Proceedings of the VLDB conference*, 2006.
- M. Boullé. A Bayes optimal approach for partitioning the values of categorical attributes. *Journal of Machine Learning Research*, 6:1431–1452, 2005.
- M. Boullé. MODL: a Bayes optimal discretization method for continuous attributes. *Machine Learning*, 65(1):131–165, 2006.
- M. Boullé. Compression-based averaging of selective naive Bayes classifiers. *Journal of Machine Learning Research*, 8:1659–1685, 2007.
- L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone. *Classification and Regression Trees*. California: Wadsworth International, 1984.
- P. Chapman, J. Clinton, R. Kerber, T. Khabaza, T. Reinartz, C. Shearer, and R. Wirth. *CRISP-DM 1.0 : step-by-step data mining guide*, 2000.
- S. Chaudhuri and R. Motwani. On sampling and relational operators. In *IEEE on Data Engineering*, 1999.
- P. S. Efraimidis and P. G. Spirakis. Weighted random sampling. Technical report, Research Academic Computer Technology Institute, 2004.
- U.M. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. From data mining to knowledge discovery: An overview. In *Advances in Knowledge Discovery and Data Mining*, pages 1–34. AAAI/MIT Press, 1996.
- R. Féraud and F. Clérot. A methodology to explain neural network classification. *Neural Networks*, 15:237–246, 2001.
- A. Gionis, P. Indyk, and R. Motwani. Similarity search in high dimensions via hashing. In *VLDB Conference*, 1999.
- I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.
- I. Guyon, S. Gunn, M. Nikraves, and L. Zadeh, editors. *Feature Extraction: Foundations And Applications*. Springer, 2006.
- G.V. Kass. An exploratory technique for investigating large quantities of categorical data. *Applied Statistics*, 29(2):119–127, 1980.

- R. Kohavi and G. John. Wrappers for feature selection. *Artificial Intelligence*, 97(1-2): 273–324, 1997.
- M. Kolonko and D. Wasch. Sequential reservoir sampling with a non-uniform distribution. Technical report, University of Clausthal, 2004.
- V. Lemaire and R. Féraud. Driven forward features selection: a comparative study on neural networks. pages 693–702, 2006.
- D. Pyle. *Data preparation for data mining*. Morgan Kaufmann Publishers, Inc. San Francisco, USA, 1999.
- J.R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.
- G. Saporta. *Probabilités analyse des données et statistique*. Technip, 1990.
- J.S. Vitter. Random sampling with a reservoir. *ACM Trans. Math. Software*, 11(1):37–57, 1985.