Modelling Complex Data by Learning which Variable to Construct

Françoise Fessant, Aurélie Le Cam, Marc Boullé, and Raphaël Féraud

Orange Labs,
2 avenue Pierre Marzin, 22307 Lannion, France
{francoise.fessant,aurelie.lecam,marc.boulle,raphael.feraud}@
orange-ftgroup.com
http://www.orange.com/en_EN/innovation/

Abstract. This paper addresses a task of variable selection which consists in choosing a subset of variables that is sufficient to predict the target label well. Here instead of trying to directly determine which variables are better, we make use of prior knowledge to learn the properties of good variables and guide the selection towards the most relevant dimensions. For this purpose we assume that a variable can be represented by a set of indicators that describe both the properties of the variable and its potential relationship to the targeting problem. This approach enables the prediction of the relevance of variables without measuring their value on the training instances. We devise a selection methodology that can efficiently search for new good variables in the presence of a huge number of variables and to dramatically reduce the number of variable measurements needed. Our algorithm is illustrated on an industrial CRM application.

Key words: Variable selection, classification, scoring, CRM

1 Introduction

Customer Relationship Management (CRM) is a key element of modern marketing strategies. The most practical way to build knowledge on customers in a CRM system is to produce scores to detect churn, propensity to subscribe to a new service, etc. A score (the output of a model) is an evaluation for all target variables to explain. The score is computed using customer records represented by a number of variables or features. Scores are then used by the information system for example to personalize the customer relationship. The rapid and robust detection of the most predictive variables can be a key factor in a marketing application.

An industrial customer platform has been developed at Orange Labs to industrialize the data mining process for marketing purpose. The platform, capable of building predictive models for datasets having a very large number of input variables (thousands) and instances (hundreds of thousands), is currently in use by Orange marketing. Its fully automated data processing machinery includes:

data preparation, model building, and model deployment. The system extracts a large number of features from a relational database, selects a subset of informative variables and efficiently builds in a few hours an accurate classifier. When the models are deployed, the platform exploits sophisticated indexing structures and parallelization in order to compute the scores of millions of customers, using the best representation. The platform allows building predictive models using two orders of magnitude more exploratory variables than the current state of the art, resulting in a dramatic improvement of performances. Performances of the in-house platform have been benchmarked in an academic context through the recent challenge KDD cup 2009 [1].

Experiments on several marketing campaigns have shown that the improvement of the quality of scoring models is strongly correlated to the number of explicative variables that can be explored. However the processing time associated with data table flattening remains the main limitation to the exploration of even larger data spaces. The variables are very expensive to compute; the evaluation times growing linearly with the number of variables. For the moment, the platform is limited to the analysis of about 20 000 variables for strong industrial time constraints. The efficient exploration of such huge spaces therefore requires the conception of an exploration technique guiding the flattening towards the most promising areas.

This paper presents a methodology for the exploration of a large space of variables consistent with the time constraints. Our idea is to estimate the predictive power of input variables without measuring them and so to avoid the flattening of all variables. A variable is characterized by a set of indicators that describe both the properties of the variable and its potential relationship to the scoring problem. The link between the indicators and the predictive importance of the variables is modelized with a subset of evaluated variables. The learned model is then used to infer the predictive importance of many new variables. Then the set of best variables can be selected for final scoring. In this way, we can explore a large set of variables while measuring only a few of them. What's more we are able to characterize the most important variables and to judge new variables.

We describe the complete methodology of exploration and its evaluation on a raw marketing campaign. The paper is organized as follows: section 2 gives an overall view of the in house Orange customer analysis platform. Section 3 details the methodology of exploration. Experimental results are presented in section 4. We conclude with some further research directions in section 5.

2 Platform Description

Two main steps of the Orange in-house customer analysis platform, data preparation and model building, are described in this section. More about the platform can be found in [2].

2.1 Data Folder

Unlike the current practice of data mining architecture, the explanatory variables are not designed and computed once in a datamart. In our platform architecture, the input data from information system are structured, and stored in a simple relational database called the data folder. The explanatory variables are constructed and selected automatically for each specific marketing project. The data folder model provides a unique view of the available input data sources, normalized according to 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 N-0 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 table contains the list of services, of usages of theses services, the call details.

The star schema offers an efficient trade-of between single table data mining and full multi-relational data mining: it has a large expressiveness, suitable for many data mining problem, and it allows efficiently build aggregated variables from secondary tables. Finally, this star schema allows to design formatted and restricted data extraction languages in order to facilitate automatic control of data extraction.

2.2 Data Extraction

The platform uses a feature construction language dedicated to the marketing domain, to build tens of thousands of features in order to create a rich data representation space. The data extraction functionality of the platform is parameterized using dedicated 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.

We focus on the construction language because it represents one of the sources of prior knowledge exploited in our methodology. A unified framework is used to write each language expression. It is composed of several successive fields (an example is given table 1). The first one is the identification of the variable (”Id”), the second is the type of the variable (”Type” whose values can be numeric or symbolic). The third is the name of the table of origin (the primary table or a secondary one). Fourth item is the name of the operator (several type of operators are used, simple selection with ”Get”, calculation with ”Mean”, ”Count” or ”Total” and more complex like date and trends). Next item ”Operand” identifies the selected field in the table. The four following items correspond to a selection expression. A selection expression is defined by a naming rule ”Sel_Id_1”, the choice of another field of the table ”Sel_Operand_1”, one or more selection values ”Sel_Value_1”, and the choice of a new operator ”Sel_TranscodingOperator_1 that can be a ranking operator or a date. The selection expression enables to specify some crosses between several fields of a given table. The language expression can contain from 1 to 4 selection expressions allowing more or less complex crosses. For example, to build the total turnover for several successive quarters for all customers, one single language expression needs to be specified (the expression is illustrated table 1). The table of origin is the secondary table “Photo”, the name of the selected field is the operand identifying the turnover “CA”. The operator working on the operand is the calculation operator “ Total’. The selection expression is defined by the choices of the other field of the table (”M_Photo”), a transcoding operator (“DiffDate”) and some values for the selection ([0,1,2] means that the total amount of CA is evaluated on the three last months stored in the data folder). The language expression generates 3 variables of numerical type (the turnover for 3 successive quarters) labelled ”CA3M_t1”, ”CA3M_t2” and ”CA3M_t3”.

It is then possible to specify up to thousands of variables to construct, using one single expression of the construction language.

Id	Type	Table	Operator	Operand	Sel_Id_1	Sel_	Sel_	Sel_Trancoding_
						Operand_1	Value_1	Operator_1
CA3M	N	Photo	Total	CA	$-t$	M_ Photo	$[0, 1, 2]$; $[3, 4, 5]$; $[6, 7, 8]$	DiffdateM

Table 1. The expression generates 3 explicative variables about the turnover for 3 successive quarters (CA3M_t1, CA3M_t2, CA3M_t3). It is composed of successive fields: Id, type of the variable (N for numerical in this case), source table name, operator, operand, selection id for variable identification, operand of selection, selection values and trancoding operator. A single expression can contain from 0 to 4 selection id, selection operands, selection values and trancoding operators according to the required complexity.

2.3 Data Preparation

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 robust and efficient variable selection method has been implemented. Explicative variables are individually evaluated by means of a supervised discretization method in the numerical case or by means of an optimal value grouping method in the categorical case. Supervised discretization [3] (or value grouping [4]) is treated as a non parametric model of conditional probability of the output variable given an input variable with the MODL approach (Minimum Optimized Description Length). The discretization is turned into a model selection problem and solved in a Bayesian way. The best discretizations and value groupings are optimized using the bottom-up greedy heuristic described in [3]. One advantage of this filter approach is that non informative variables are discretized in one single interval and can thus be reliably discarded. This approach also quantitatively evaluates the predictive importance of each variable for the target.

2.4 Modelling

The orange in house platform uses the Khiops scoring tool which implements an extension of the naives Bayes classifier (including model averaging) called Selective Nave Bayes classifier. The system has no hyper-parameter to adjust. The tool is designed for the management of large datasets, with hundreds of thousands of instances and tens of thousands of variables, and was successfully evaluated in international data mining challenges. Khiops can be downloaded here: http://www.khiops.com/. Once learned, the model is finally deployed to produce scores for all instances on all the explanatory variables.

3 Predicting the Relevance of a Variable

3.1 Related Work

Our problem can be seen as a problem of variable selection. Classical variable selection task is to choose a small subset of variables that is sufficient to predict the target well. The main motivations for variable selection are computation complexity, reduction of the cost of measurements, improving classification accuracy or problem understanding [5]. The main approaches studied in the literature are filter and wrapper [6]. 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 [7] are restricted to the modelling phase of data mining, as a post-optimization of a classifier. Filter methods are better suited for the data preparation phase, since they are time efficient and can be combined with any data modelling approach.

Classical methods of variable selection tell us which variables are better, they don't tell us what characterizes these variables or how to judge new variables which were not measured in the training data. On the basis of these observations Krupka [8] has recently developed another approach to variable selection. Instead of selecting a set of better variables out of a given set, his algorithm learns the relation between some descriptors coming from prior knowledge on initial data and the variable usefulness. This in turn enables him to predict the quality of unseen variables. The scenario is based on an extension of Recursive Feature Elimination [9], a wrapper selection method for linear SVM. Subsets of variables with poor usefulness are successively removed with a recursive process. Other ideas about the exploitation of prior knowledge about relevance of variables can be found in the literature. For instance, [10] performs transfer learning across tasks, acquiring prior knowledge on one dataset and using it as partial supervision on others. [11] is another example of transfer learning. Our work is based on an idea similar to [8] that consists in exploiting prior knowledge we have on initial variables and linking it to variable relevance. The modelization is completely based on the Khiops tool.

3.2 Acquisition of Prior Knowledge on Variables

As introduced section 2, the platform allows the generation of many variables with very few language expressions. The definition of an expression is composed of several choices: table, variable, operators, operands, values, ... and specifications for the exploitation of the expression, like id for labelling the variable. The language used for the construction of the variables provides the first source of prior knowledge we want to exploit. The initial data are stored in a data folder and this data folder is another source of prior information. For example, we know for a categorical variable details about its modalities (number, frequency) and for a numerical variable the spread of values.

List of descriptors Each variable has been described by a set of descriptors from these two sources of knowledge. 15 descriptors have been directly retained from the structure of construction of the variable or derived from it:

- Type of the variable (a variable can be categorical or numerical),
- Table name (one of the table of the data folder),
- Operator (the name of the calculation operator: Get, Count, Mean, Trend, ...)
- Type of operator (an operator can be a simple selection or more complex: calculation, date, trend or count),
- Flag for the presence or absence of an operand (yes or no),
- Operand name (the name of one field of the selected table),
- Total number of transcoding operators in the expression (examples of transcoding operators: WeekDay, Diffdate, HourNumber, AscendingRanking, ...),
- Transcoding operator names (vector of 2 dimensions, in our applicative context an expression can have at most 2 items filled),

- Number of transcoding operator in each type (vector of 2 dimensions, a transcoding operator can be a date or a ranking),
- Number of selection operands (a selection operand is a field item of the selected table),
- Names of selection operands (vector of 4 dimensions, in our construction scheme a language expression can have up to 4 items filled in the selection expression),
- Length of the language expression (total number of items in the language expression),
- Flag for the complexity of the expression (yes or no, an expression is considered as complex if at least a part of a selection expression is filled in),
- Number of selection Id (a selection Id is used to label the variable),
- Number of selection values in each type (vector of 6 dimensions, a value can be a single numerical or categorical value, an interval of numerical values, a group of numerical or categorical values, a null value).

The 5 descriptors retained from the initial data in data folder are:

- Operand type (an operand can be numerical, categorical , a date or a time value),
- Number of operands in the table,
- Number of modalities for a categorical operand,
- Entropy for a numerical operand,
- Ratio between the interquartile interval and the median for a numerical operand.

Finally, prior knowledge on an explicative variable is represented by a vector of 30 dimensions.

3.3 Model of Variable Importance

We now define a new supervised problem. The original variables are the instances. The descriptors listed above become the new variables. The target is the predictive importance evaluated by the scoring model. As recalled section 2.3, Khiops analyses each variable independently for the target and return a value that is directly its predictive importance. Khiops is used once again as a classification model to find the required mapping from descriptors to predictive importance. The algorithmic protocol is decomposed into the learning and test steps:

Learning Step: We are able to build a set of N variables from a set of P feature construction expressions. We assume that we evaluate only a subset of these N variables with the scoring platform (it means that only these variables are flattening and a predictive importance is available for each of them). The descriptors associated to this subset of variables correspond to our learning set. We use it to learn the relation between the descriptor values and the variable importance. The problem we learn is not the exact prediction of the importance value but the class of importance (i.e. if the predictive importance value is null (not important) or positive (important)).

Test Step: Based on the previous modelization, the goal is now to generalize to unseen variables. We predict the importance class for the instances of the test set represented by the descriptors of the whole variables including variables that were not part of the training set. This in turn enables us to choose the most relevant variables for the final scoring. The process of variable importance prediction can be summarized as follows:

- Variable and descriptor sets constitution

  Variable set: build variables from a limited number of language expressions
  Descriptor set: extract descriptors for each variable
    15 descriptors based on the language framework
    5 descriptors based on the data stored in the data folder

- Learning of the model of importance
- Generalization on all the constructed variables

  Selection of the most important variables

4 Experimental Validation

We report in this section practical experiments that have been made on a raw marketing campaign.

4.1 Data Description

For the evaluation, the platform is supplied with data collected on a sample of 30000 customers. The information comes from decisional applications of Orange Company. The goal of the task presented here is to prevent a customer to switch ADSL provider. For this problem we have $24,3\%$ of positive instances. The feature construction language is used to generate 20000 initial explicative variables from 600 feature construction expressions (an example of such expression is given table 1).

4.2 Evaluation Process

The final evaluation concerns the scores produced with the platform. We compared the scores for several sizes of subsets in the model of predictive importance (it means that only the variables corresponding to these subsets are initially flattened and evaluated with the platform). The complete algorithmic protocol is as follows:

- Learning set constitution

```
    Repeat
      Random selection of a language expression
        Random selection of a variable among those generated by the expression
    Until the expected number of variables is reached
    Evaluation of the importance associated to variables with the scoring platform
    Building of the set of descriptors for the set of variables
– Learning of the model of importance

– Generalization on all the constructed variables
     Selection of the most important variables
– Final scoring with the selected variables
    Scoring evaluation.
```

4.3 Results

We successively experimented with a sample of 2, 5, 10, 20 and 40 percent of the initial explicative variables. In other words, the model of importance has been built with respectively 400, 1000, 2000, 4000 and 8000 instances (the instances being selected as described section 4.2). The predictive model is evaluated using the area under the ROC curve (AUC) [12] (the higher the criteria, the better, with 1.00 indicating perfect performance).

Table 2 shows for each sample, the number of variables evaluated for the constitution of the learning step, the time of flattening, the AUC of the classifier on the test set, the number of variables that have been classified with a positive predictive importance after generalization and the number of variables really important among them. 1072 variables have been labelled as important when the scoring has been achieved directly with the flattening of all the initial explicative variables. A evaluated variable is tagged as really important if it belongs to this set. 70% of the users are used for the modelization steps, the remaining 30% are kept for the final scoring evaluation.

We observed that less than 4% of the whole variables is considered as important for the targeting by the model. This number regularly increases with the size of the subset used in the learning step. A detailed analysis of the model of importance can help us to characterize good variables (for instance, the descriptors with high level in the model are the name of the operand, the names of the first and second selection operand in the expression and the name of the table).

Only the variables predicted as important are considered now and flattened for final scoring.

We compared the scores produced for the 5 sets of variables predicted as important to those given by the current operational model. The current model requires the direct flattening of 20000 explicative variables.

The performance of a model is measured with the cumulative gain curve. It is a graphical representation of the advantage of using a predictive model to choose

Sample rate	size of the learning set for importance prediction	flattening time (m)	AUC	nb of variables classified as important	nb of variables really important
2 %	400	35	0.858	284	177
5 %	1000	51	0.850	513	413
10 %	2000	67	0.905	595	439
20 %	4000	121	0.908	604	450
40 %	8000	210	0.913	775	513

Table 2. Sample parameters for learning the model of importance, flattening time (in minutes), AUC of the classifier on the test set, number of variables classified as important and number of really important variables. It takes 375 minutes to flatten the initial set of 20000 explicative variables.

which customers to contact. The x-axis gives the proportion of the population with the best probability to correspond to the target, according to the model. The y-axis gives the percentage of the targeted population reached. The curves are plotted on Figure 1. The diagonal represents the performance of a random model. If we target 20% of the population with the random model, we are able to reach 20% of the fragile customers. With the current model, when 20% of the population is contacted, 40% of the fragile customers is reached.

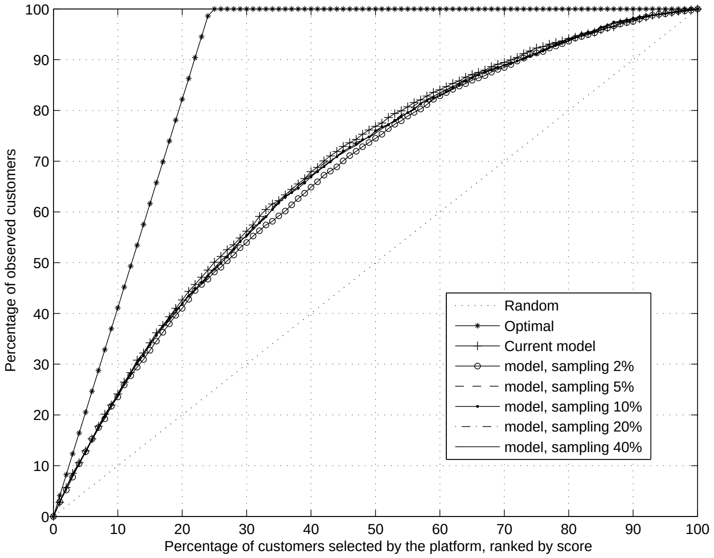

Fig. 1. Lift curves of scoring models.

The curves corresponding to the sampling rates of 5%, 10%, 20% and 40% are almost confused and this remains true for the entire cumulative gain curve. The performance slightly decreases for the sampling at 2%. Numerical results in table 3 complete the previous observations. We give the AUC of the different scoring models. The lowest sampling rate excepted, the scoring based on the variable selection scenario has led to the same scoring accuracy than the actual model. For instance, a sampling of 5% of the initial variables means that 1000 variables among 20000 are first flattened in order to build the model of predictive importance. At the end of the generalization, 513 variables considered as important are retained. In the end the complete scoring process required the evaluation of about 1500 variables. Therefore we can conclude that a reduction of a factor 12 of the number of evaluated variables has been achieved without damage on the final scoring.

model (sample rate)	AUC
Current model	0.744
2 %	0.728
5 %	0.735
10 %	0.739
20 %	0.738
40 %	0.740

Table 3. Final scoring model performances (AUC).

The experimental results confirmed the interest of the approach. We obtained similar scoring performances to the actual model with a significant reduction of measurements. A consequence is an important saving of time for the global scoring process. Another point with the method is that we are able to characterize the properties of good variables. An in depth analysis of the 20 best variables kept by the targeting models shows that they share 40% of similar variables with the current model. We can notice that efficient scoring can be achieved with several combinations of variables.

5 Conclusion

We have described in this paper a methodology of variable selection whose main idea is to take benefit from prior knowledge on variables to guide the exploration of the input space towards the most promising areas. The approach consists in predicting the quality of variables with measuring few of them. A variable is described by a set of indicators and the link between these indicators and the predictive importance of the variable is modelized. The model is then used to predict the importance of new variables. Only the variables predicted as important are retained and evaluated for final scoring, the other being discarded.

The result is a dramatically reduction of the number of variable measurement needed for a similar scoring performance. With this approach, for a given number of variables we can explore more quickly or explore more variables in a fixed duration.

The validity of the approach has been demonstrated on a raw marketing campaign for several thousands of variables. This preliminary work needs to be extended. The exploration of even larger input spaces raises the question of overfitting and the risk that a variable becomes informative by accident. A solution could be a regularization procedure to penalize variables whose computational cost is high. Another research perspective is to combine our methodology with another learning method. A promising example is discussed in [13] where variable selection is formalized as a reinforcement learning problem.

References

1. Guyon, I., Lemaire, V., Boullé, M., Dror, G., Vogel, D.: Analysis of the kdd cup 2009: Fast scoring on a large orange customer database. Journal of Machine Learning Research: Workshop and Conference Proceedings **7** (2010) 1–22
2. Féraud, F., Boullé, M., Clérot, F., Fessant, F., Lemaire, V.: The orange customer analysis platform. In Perner, P., Ahlemeyer-Stubbe, A., eds.: Proceedings of the 10^{th} industrial conference on data mining, Springer Verlag (2010)
3. Boullé, M.: MODL: a Bayes optimal discretization method for continuous attributes. Machine Learning **65**(1) (2006) 131–165
4. Boullé, M.: A Bayes optimal approach for partitioning the values of categorical attributes. Journal of Machine Learning Research **6** (2005) 1431–1452
5. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. Journal of Machine Learning Research **3** (2003) 1157–1182
6. Kohavi, R., John, G.: Wrappers for feature selection. Artificial Intelligence **97**(1-2) (1997) 273–324
7. Féraud, R., Clérot, F.: A methodology to explain neural network classification. Neural Networks **15** (2001) 237–246
8. Krupka, E., Navot, A., Tishby, N.: Learning to select features using their properties. Journal of Machine Learning Research **9** (2008) 2349–2376
9. Guyon, I., J. Weston, S.B., Vapnik2, V.: Gene selection for cancer classification using support vector machines. Machine Learning **46**(1-3) (2002) 389–422
10. Lee, S., Chatalbashev, V., Vickrey, D., Koller, D.: Learning a meta-level prior for feature relevance from multiple related tasks. (2007) 489–496
11. Helleputte, T., Dupont, P.: Partially supervised feature selection with regularized linear models. In Bottou, L., Littman, M., eds.: Proceedings of the 26th International Conference on Machine Learning, Montreal, Omnipress (June 2009) 409–416
12. Fawcett, T.: ROC graphs: Notes and practical considerations for researchers. Technical Report HPL-2003-4, HP Laboratories (2003)
13. Gaudel, R., Sebag, M.: Feature selection as a one-player game. In: Proceedings of the second NIPS Workshop on Optimization for Machine Learning (OPT 2009). (2009)