# Managing the Process of Segmentation on the Mobile Phone Subscribers

http://dx.doi.org/10.3991/ijes.v3i2.4491

Rodrigue Carlos Nana Mbinkeu[1,2], Domenico Beneventano[2]
[1]University of Yaoundé I, Cameroon
[2]University of Modena, Italy

*Abstract*—**Most telecommunications providers possess a remarkable amount of data about their subscribers. The knowledge that we would discover in the database of telecommunications providers is vital to understanding the behavior of subscribers. We talk about subscribers segmentation. The segmentation will identify and select the subscribers most likely to respond favorably to offers. Our paper proposes a set of techniques to analyze and design tools that manages the process of data acquisition, data cleaning and selection of the segmentation algorithm.**

*Index Terms*—**Data mining, Segmentation, Subscribers, Mobile Phone, Data Cleaning, Data Acquisition, DataBases, Java API.**

## I. INTRODUCTION

The telecommunication operators generate and store a tremendous amount of data. These data include call detail data, network data, the state of hardware and software components in the network, and customer data [10]. The knowledge about mobile customers becomes one of the important assets in making business decisions.

Segmentation is very important in the field of direct marketing [2]. The segmentation process in the company has to take into account knowledge of business people about the data, the features that might be used for clustering and the features of customers that might be used for profiling. Behavioral segmentation is a process of finding the groups of customers with similar behavioral pattern. It is one of the key data mining tasks for marketing departments of telecommunication companies [2]. The knowledge gathered can be used for different purposes: call individual clients, make offers tailored to target groups, predict customer behavior like as a churn [10].

Our approach provides automatic management of acquisition phases and data cleaning, the choice of the segmentation algorithm, visualization of segments of interest and statistics. In this paper we will focus on telecommunication industry. However, the methods that we will present are also valid for other industries.

This paper is organized as follows: Section 2 presents the data mining tools. In Section 3, we present how to build the conceptual level data. Then, we work on data quality in section 4. Finally, we describe how our prototype works in section 5.

## II. DATA MINING TOOLS

Data mining would not exist without tools. The data mining software programs are specialized in analyzing and extracting knowledge from computerized data. Currently the most used tools are: SPSS, RapidMiner, SAS, Excel, R, KXEN, Weka, Matlab, KNIME, Microsoft SQL Server, Oracle and STATISTICA DM [7]. Most of these tools do not allow an easy and intuitive use by persons not expert in knowledge discovery data.

Our techniques are designed to automate the process of segmentation. These techniques will be used to design tools that enable non-experts to do the segmentation. As part of our study and the implementation of a prototype, we used the data warehouse Oracle and Java Data Mining Server [1]. Tasks related to data acquisition, data cleaning and the choice of segmentation algorithm are defined by the domain expert once. Scheduled tasks are stored in an XML file. Our techniques use the CRISP-DM methodology [5, 4] and can be implemented with different databases.

## III. DATA DOMAIN UNDERSTANDING

Data Domain understanding is one of the first steps in real-world applications of Knowledge Discovery in Databases (KDD). Preprocessing the data properly and finding an appropriate representation is crucial for a successful application of data mining algorithms [3]. In our case, we need to formally represent the concepts from the domain understanding and link them with the relations of database. Euler and Scholz [6] present a method using ontology for modeling abstract data models and their interconnection with the real data in database. We were inspired by their work to build our data model in the formalism that we present to the next subsection.

### *A. Building the conceptual level data*

This process deal with the mapping of low-level data into other forms those are more compact, abstract and useful (ontology level). This is achieved by creating higher level of data abstraction and the next step consists to model the process of collecting and cleaning data. Well-known formalisms for data abstraction like entity relationship diagrams allow to model concepts and some of their properties intuitively. Once the schema of higher level has been established it can be considered as an interface for inserting, updating and deleting objects.

For example, at the ontology level: the concept subscribers and its features allow us to describe the data more abstract. This approach allows us to obtain a new view of the existing database. In this case, preprocessing consists of loading the data from various sources and transforming it into a table where each row corresponds to a subscriber and columns correspond to the selected features of subscribers. The principal advantage of this two-level data model is that all the data processing will be described in terms of the ontology level, which allows to re-use the complete description on a new database by simply changing the initial mapping [6]. The concepts obtained represent domain ontology and offer a simplified and more understandable data of database. In the following lines, we propose a general procedure for calculating the instances of features associated with concepts:

- **The Time Dimension:** We study the behavior of subscribers over a specific period. It can be defined in day, week, month ... etc. For example, we could use the month to set our smallest unit of time. Relations of database typically have each column defining the date of the events it store. The more granular tables give the day and time of the event, other more aggregated rather give the month or year. We will specify a sql expression on each table to obtain the month and year of events as default value of time.
- **Calculating the instance of concept:** We may want to have several concepts that use the same database table (using the same or different sets of columns as features). These concepts can serve different purposes in the KDD application in question. We link the features of the concept and tables from which their values are calculated. We specify for each feature of the concept:
  - the table (tables) from which (whom) its value is calculated;
  - an sql expression which gives the feature value from one or more columns of the table (or tables).

Several features (attributes) of the concept can be linked to the same database table. The data model defined in the formalism presented above has features that are related to the tables of database. We propose a procedure to calculate the instance of subscribers concept. It runs as follows:

- Create a concept and associated with the corresponding features;
- Create queries (sql expressions) associated with features on database tables related;
- Run the queries obtained. Note that these queries can be executed in parallel;
- Join the results for all values of features in the view created (view represents a concept).

The data model defined in the formalism above is flexible. Features can be added or removed from the concept, the link between the features and tables can be changed and these changes will automatically be considered at the next execution of the process of data collection. The result is a weak coupling between the conceptual level and the database. This weak coupling meets the need of users to extend the concept (taking into account new features) during future segmentations.

## IV. PROCESSING DATA QUALITY

Quality of data collected to create instances of concepts must be analyzed. The data will be cleaned before being submitted to the Data Mining Server. It will take care of their separation in segments (clusters). The reduction of the concept consists to remove some features which have no impact on the calculation of the segments (the reduction procedure is illustrated in 4.1). The statistics will be calculated on the remaining features.

- **Treatment of null value:** NULL values indicate missing values. Features may have null values (no value). Missing values in these cases are generally due to non use of certain services by subscribers. By default, we will replace the null values with 0 in the case of numerical value, except in the case where the user specify one of the following treatments for a feature:
  - deletion: tuples with NULL values for this feature will be discarded for future operations of the process;
  - Substitution: NULL values will be replaced with an actual value specified by the user;
- **Treatment of outliers:** Outliers are values that lie beyond the scope of the value range of a field. For example, the presence of outliers may be due by subscribers with extreme use of the phone; or a phone used by a community (call box) or a fraud situation. These situations do not correspond to normal use of the phone and they can influence the construction of the segments. However, in some particular cases, we could keep them. In fact the outliers are interesting for modeling abnormal behaviors [9]. When we want to model normal behavior, it would be important to remove outliers. In this case, the user chooses an algorithm that detects outliers [9]. The following basic tasks must be performed:
  - Specify a confidence interval for each feature;
  - Tuples whose feature values are outside the defined interval will be discarded.

### A. *Reduction of the concept*

The reduction operation will help to select the relevant features for segmentation. To do this, it will automatically perform the tasks we have described in earlier phases. The reduction procedure is as follows:

- Remove the features whose values do not vary enough;
- Calculate the variance of the different features after normalization (i.e. after converting all values between 0 and 1);
- Remove the features whose variance is below a threshold set by the user. For example, it may happen that the number of voice messages sent by users does not vary enough because the service is used by only 3 per 100 of the actual population of subscribers. Similar observations can be verified on other features.
- Separate features strongly correlated
  - Group features in pairs;
  - Calculate the correlation coefficient for each pair;
  - Rank pairs in order of decreasing correlation;
  - Establish a ranking of features according to the largest correlation with each another feature below the threshold (default 0.9);
  - Browse couples with a correlation above the threshold in each pair and remove one of two features (if not already removed);

### B. Attribute Importance

Attribute Importance provides a solution for improving the speed and possibly the accuracy of classification models built on instance of concept with a large number of features. The time required to build segmentation increases with the number of features. Attribute Importance identifies a proper subset of the features that are most relevant to predicting the target variable [8]; for example the target variable may be the income generated by subscribers. Model building can proceed using the selected features only.

In our prototype, we apply the principle of Minimum description length with the algorithm attribute importance specified in the Java Data Mining Server [1].The features are ranked according to their relevance in predicting a target variable and the less relevant will be removed.

Data obtained after the steps described above are ready to be processed by different segmentation algorithms.

## V. OUR PROTOTYPE

We implemented a Web platform for building a subscriber segmentation using the approach described above. We show in the figure 1, the principal items of menu. Different algorithms [3] (like as K-means, O-cluster, etc...) can be used in the web platform for segementation. In the figure 2, we show that the user can define the concept as a view and interconnect it with database tables. Each concept is associated with the set of features (attributes). User specifies for each feature the query expression to select data in the database tables.

The system guides the user for cleaning data. Then, the user must define a set of operations that allows selecting the relevant attributes (variance, correlation).

All these basic tasks defined by the expert are stored in an xml file of the web platform. This has the advantage of automating the process for the future segmentations.

As part of our collaboration with the mobile phone company Orange, our web platform has been able to handle the segmentation process on a set of seven million subscribers.

## VI. CONCLUSION

In this paper, we presented a set of techniques that improve the analysis and the design of tools for managing the process of segmentation. The use of these techniques can automate the process of segmentation for non-expert users. A prototype is implemented and is currently used by Orange Cameroon.

As perspective, we plan to extend this work by considering other sectors of industry and by adding new functionalities.

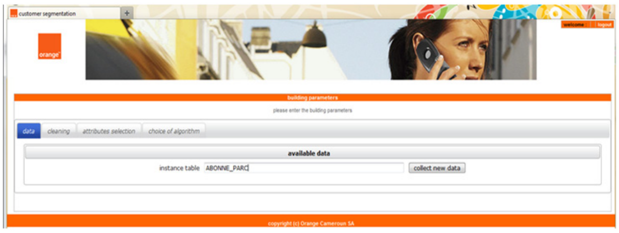
## REFERENCES

[1] Satoshi Oracle data mining application developers guide 11g and java data mining api.

[2] Edmunds communications group. the importance of data mining and segmentation indirect marketing. 2010.

[3] Ayaz Ali and Chen. Scalable clustering algorithms. 2004.

[4] A. Azevedo. Semma and crisp-dm: A parallel overview. 2008.

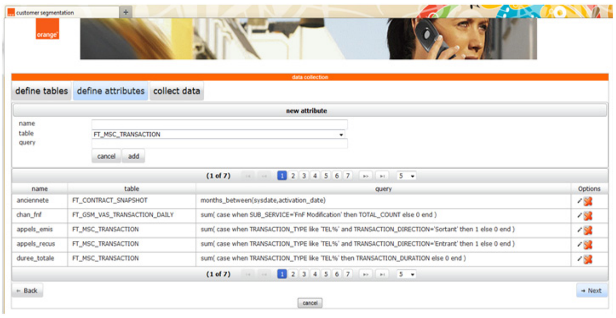


please enter the building parameters

available data

instance table ABONNE_PARC

Figure 1. General Menu



new attribute

name

table FT_MSC_TRANSACTION

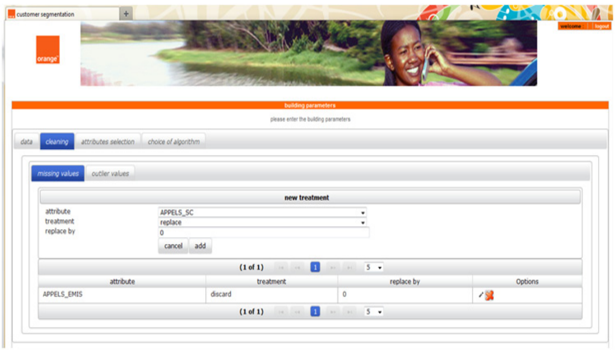
query

(1 of 7) 1 2 3 4 5 6 7 5

| name | table | query | Options |
|---|---|---|---|
| anciennete | FT_CONTRACT_SNAPSHOT | months_between(sysdate,activation_date) | |
| chan_fnf | FT_GSM_VAS_TRANSACTION_DAILY | sum( case when SUB_SERVICE='FnF Modification' then TOTAL_COUNT else 0 end ) | |
| appels_emis | FT_MSC_TRANSACTION | sum( case when TRANSACTION_TYPE like 'TEL%' and TRANSACTION_DIRECTION='Sortant' then 1 else 0 end ) | |
| appels_recus | FT_MSC_TRANSACTION | sum( case when TRANSACTION_TYPE like 'TEL%' and TRANSACTION_DIRECTION='Entrant' then 1 else 0 end ) | |
| duree_totale | FT_MSC_TRANSACTION | sum( case when TRANSACTION_TYPE like 'TEL%' then TRANSACTION_DURATION else 0 end ) | |

(1 of 7) 1 2 3 4 5 6 7 5

Figure 2. Calculating the instance of concept



please enter the building parameters

new treatment

attribute APPELS_SC

treatment replace

replace by 0

(1 of 1) 1 5

| attribute | treatment | replace by | Options |
|---|---|---|---|
| APPELS_EMIS | discard | 0 | |

(1 of 1) 1 5

Figure 3. Data cleaning

[5] P. Chapman. Crisp-DM 1.0 step-by-step Data Mining Guide. 2000.

[6] T. Euler and M. Scholz. Using ontologies in a kdd workbench. Workshop on Knowledge Discovery and Ontologies at ECML-PKDD, 2004.

[7] K. Rexer and P. Gearan. Data miner survey summary. 2010.

[8] I. Kononenko. Evaluating the quality of attribute. 2005.

[9] V. Saltenis. Outlier detection based on the distribution of distances between data points.Journal Informatica, 2005.

[10] G. M. Weiss. Data mining in the telecommunications industry. In Data Mining and Knowledge Discovery Handbook, pages 1189–1201, 2005.

## AUTHORS

**Nana Mbinkeu Rodrigue Carlos** is a Senior Lecturer at Department of software Engineering of the National Advanced School of Engineering, P.O Box 8390 ENSP, University of Yaoundé I, Cameroon (nanambinkeu@gmail.com).

**Domenico Beneventano** is an Associate Professor at University of Modena and Reggio Emilia in Italy (domenico.beneventano@unimore.it).

Submitted 24 February 2015. Published as resubmitted by the authors 12 May 2015.