AUTOMATIC CONTENT TARGETING ON MOBILE PHONES

Giovanni Giuffrida
Dip. di Matematica e Informatica
Università di Catania
Viale Andrea Doria, 6 – 95125, Catania
+39 095 7383051
ggiuffrida@dmi.unict.it

Catarina Sismeiro
Tanaka Business School
Imperial College
South Kensington Campus
London SW7 2AZ
+44 (0) 2075949108
sismeiro@imperial.ac.uk

Giuseppe Tribulato
Dip. di Matematica e Informatica
Università di Catania
Viale Andrea Doria, 6 – 95125, Catania
+39 095 7383051
tribulato@dmi.unict.it

ABSTRACT
The mobile phone industry has reached a saturation point. With low growth rates and fewer new customers available to acquire, competition among mobile operators is now focused on attracting competitors' customers. This leads to a significant downward price pressure, the inability by mobile phone providers in deriving reasonable returns from basic telephony services, and an increasing reliance on value added services (VAS) for revenue growth. There are today thousands of such services available for companies to sell to their customers daily. These services include, for example, the provision of sports information, ring-tones, personalized news, weather forecast, and financial trends. Because of the many possible offers, and of the limited contact opportunities (operators tend to cap the number of commercial messages sent to their users and phones have limited-size screens), data mining can play an important role in optimizing message targeting. In this paper we describe our experience in developing a successful automatic system to target users with the most relevant offers. We describe the proposed data mining methods and report on their performance. In addition, we discuss several experiments we implemented on live data. These experiments have been useful to tailor our approach to the specific characteristics of the market under study. We believe this is a very interesting domain for data miners though it is still fairly unexplored. This is despite the availability of very large and detailed logs of customer activity.

Keywords
Mobile customers targeting, Mobile advertising, Data mining, Clustering.

1. INTRODUCTION
In most developed countries, the mobile phone market is becoming increasingly saturated and competitive. Mobile phone penetration is over 100% in several European countries, and first-time customers (new users that enter the market and expand the business) are practically inexistent [14]. Because markets are stagnant, mobile businesses are now focusing on convincing competitors' customers to switch to their networks. The main factor that influences customers' choice of operator is the availability of a more convenient telephony rate plan. Whereas lower telephony rates tend to reduce revenues for the company, and may even produce a (tolerated) loss, when properly managed value added services tend to produce additional revenues (and significant profits). As a result, mobile operators rely increasingly on price competition for customer acquisition, and on VAS offerings for revenue expansion. For all these reasons, the quality and variety of these new services, and the management of the VAS offerings, are now crucial for the success of mobile operators.

Because companies recognize that these services are essential for their profitability, mobile phone operators are becoming very creative in developing and proposing new offerings to their customers. For example, mobile phone users can receive real-time weather forecasts, real-time stock trends, sports information, general news, and location based services. Users can personalize their devices by downloading ring-tones and wallpapers. Entertainment is also easily accessible with increasing number of songs, games, and videos available for download. In addition, with the latest handsets, users in some countries can already browse the Internet, check their emails, and watch TV shows as often as they desire.

The growing number of possible services to offer, the limited size of mobile phone screens, and the risk of alienating and overwhelming users if too many messages are sent, makes the selection of user-specific content key to success. In other words, it is essential for the service provider to understand the customer's needs and interests in order to select the right services to promote.

A significant advantage of mobile operators in managing VAS is that current infrastructures keep detailed logs of customer interaction with the offered services. These logs keep track of all the messages and offers sent to a customer, and of the corresponding feedback (e.g., whether the customer opened a message, viewed a page, bought a video, or clicked on a link). The information contained in these logs can then be used by an automated system to aid message selection and customer targeting.

In this paper we discuss our experience in developing a clustering based tool to target mobile customers. The remainder of the paper is structured as follows. First, we present the challenges faced by

EDBT'07, March 25–30, 2008, Nantes, France.

mobile phone VAS based on text or multimedia messages, discuss the details of our algorithm, the infrastructure we relied upon, and our results. We conclude with future research directions.

2. TARGETING CHALLENGES IN MOBILE PHONE VAS BASED ON TEXT OR MULTIEDIA MESSAGES

Mobile phone VAS include services based on text (SMS) and/or multimedia (MMS) messages sent periodically to customers. These messages contain typically one or more commercial offers that invite users to subscribe or purchase services and to download digital products (e.g., ring-tones, TV shows, video clips). In general these commercial activities require either an opt-in or opt-out from the customer, though such requirements vary from country to country.

From the mobile operators' point of view, providing these services can be very cost effective because companies can reach millions of potential buyers very efficiently; the cost of operations is often dominated by the one-time investment on the message sending infrastructure and, subsequently, each message can be sent at zero (or close to zero) marginal cost. Profit potential is then very high and content is usually provided by third parties with whom the telecom companies share revenue. A system that optimizes customer targeting can provide a profitability boost. Such a system would send to each customer the message that maximizes expected revenue, while trying to satisfy customer needs and desires as much as possible.

Despite the significant benefits companies can attain from good targeting and message selection in this context, these activities present serious challenges. These challenges lead also to specific requirements on a mobile targeting system and its architecture. Below we will discuss such challenges and in the following sections we present the details of the targeting and knowledge discovery system that we propose to deal with each one of these challenges.

2.1 Massive number of offers

One first challenge relates to the extremely high number of alternative offers available to be sent (offers that, in turn, need to be tested). For example, the telecom company where our systems were implemented had more than 50 thousand possible products to advertise at any moment, and the list never stopped growing (e.g., in our application the content catalogue grows by twenty to thirty new items a day, and this growth rate is not likely to be reduced). This massive number of offers to be tested and learn on poses some difficulties in terms of knowledge discovery, specially when coupled with the limited contact opportunities and the infrastructure limitations (two additional challenges discussed next).

2.2 Limited contact opportunities per customer

A second challenge relates to the small number of contact opportunities per customer. Even though companies in this industry have millions of customers to contact, receiving too many commercial messages a day increases the likelihood that a customer will cancel a service, or switch operator, due to annoyance (in fact, few well-targeted messages are more effective than many generic ones [3]). Hence, operators do not want to fill customers' inboxes with too many messages and tend to limit the number of commercial messages sent to each customer. Considering the limited screen size of users' handsets, each person can only be exposed to no more than a very small fraction of all possible offers.

In our application, company policies restrict the number of daily messages per customer to one. Hence, the system sends one *single* daily message to few million customers. Each message contains a variable number of offers (one to four), and each offer advertises a specific product or service that can be purchased directly from the mobile phone with few clicks. Notice that this model is very different from a supermarket-like context where, on a single visit and in a short period of time, a customer is exposed to thousands of different products and commercial offers (though a consumer might buy only one item per category, supermarkets carry hundreds of alternatives in each category).

2.3 Infrastructure limitations

A third challenge associated with the targeting and knowledge discovery in our context relates to infrastructure limitations. One limitation regards the different messages that can be sent daily. Typically, message delivery systems can cope with reaching millions of customers a day as long as the number of *different* messages sent to users is not high. This means that we cannot associate each customer with a fully personalized message, but we can send no more than one hundred different messages, each one to thousands of individuals. As a result, full customization (one customized message per individual) is not feasible. However, we are still able to contact the millions of users we want to reach if individuals are grouped in a meaningful way (e.g., in clusters based on previous response to offers).

2.4 Content categorization

A final challenge relates to the different categorization of VAS offers received from each content provider with whom the telecommunication company contracts. Because each producer provides his own content, created independently, each producer has also developed his unique categorization schema. For instance, a java game from producer A might be classified in a category called "Entertainment." A similar java game from producer B could instead be classified by that producer as "Online Games." Hence, the offers coming from multiple producers can be assigned to categories with very different names and with a very different breadth (e.g., "Entertainment" as a category will include many other types of offers, not only online games). The differences in name and scope of vendor-specific categories poses an optimization and clustering problem. Content category could be a very powerful predictor. Despite this potential, given the way the category information is currently collected by mobile phone companies, this variable introduces mostly noise into the analysis.

Our proposed approach can cope with each one of these challenges in an effective way. Next we present the steps of our basic approach designed that deal with these challenges.

3. CUSTOMER CLUSTERING

Our general approach to solve the challenges described above starts with the efficient clustering of mobile users: before we attempt to learn the performance of the different offers, we group users into homogenous clusters. This grouping will facilitate message delivery and learning. Because we cannot target individual consumers with a fully customized offer, we will be targeting clusters of consumers and send to all the users in a cluster the same message. This is something the delivery system can handle efficiently.

User clustering is best achieved through the use of clustering algorithms that group data points based on user-defined *metrics*. In our specific case we decided not to use user demographic information in our metrics for two reasons. First, we detected excessive noise in the demographics data, which is also characterized by a significant number of missing information. Second, extensive research in the field of marketing, both in online and bricks-and-mortar environments, has concluded that standard demographics information, similar to the one available to us (e.g., gender and age), is rarely predictive of consumer decision making. Instead, past purchase and consumption behavior provides far better predictions of future purchases and consumption (see for example [8] and [13]).

As a result, we will rely on user *behavior*, in the form of their purchase histories, to perform the clustering analysis. In our context, we will assume that two customers are *similar* if they buy similar content over time or, more precisely, if they shop in similar categories in a similar proportion (we will explain in Section 4 how we solve the problem of diverse categorizations).

Next, we will explain in more detail how this clustering is performed and implemented.

3.1 Clustering metrics

Let u be a customer. We define the purchase history p(u) of user u as a vector describing the user's past purchases. For instance, p(u)= $[i_1,i_2,\ldots,i_n]$ means that the user u bought i_1 items of category c_1, i_2 items of c_2, and so on.

Choosing the right metrics for the clustering is crucial to obtain a good performance. The simple Euclidean metrics L_2 does not apply well to the similarity concept between users discussed above. Suppose, for instance, that we have two categories (c_1 and c_2) and three customers (u_1, u_2 and u_3). Assume that p(u_1) = [3, 1], p(u_2) = [1, 0] and p(u_3) = [0, 1]. If we were to apply L_2 metrics we would conclude that u_2 and u_3 are close to each other, even though these customers did not purchase from any common category. In contrast, for our application, we would want u_1 and u_2 to be closer as they purchased in the same categories (even if they exhibit different quantities in each category).

Hence, we based our clustering metrics on the dot product of the purchase histories, which is defined as follows:

$$D_p(a,b) = 1 - \sum a_i' b_i'$$

where the two sets *a'* e *b'* are the normalized version of vectors a and b so that

$$\|a_i'\| = \|b_i'\| = 1 .$$

We normalize our vectors using the *tfn* schema known as "*normalized term frequency – inverse document frequency*" (please see [11] for other normalization schemas).

If one applies this metric, we obtain a greater affinity between user u_1 and user u_2, which is what we desire.

We cluster users daily to account for new purchase history information and to guarantee that if any significant changes occur, these can be effectively captured. This operation has to be completed in few hours between the arrival of the logs and the scheduling time.

Because of its good scalability and speed, given the characteristics of our application, we have adopted the Spherical k-means algorithm presented in [5] for user clustering. This is a particular version of the historical k-means [12] and is based on the dot-product metrics discussed above. Though a faster version of the Spherical k-means has been proposed in [6], we have opted to use the original spherical k-means for two reasons. First, its execution takes $O(nki)$ (where i is the number of iterations), making it suitable for the analysis of millions of data points in a short period of time without a significant memory burden. Second, to memorize the execution state between iterations we only need to save the set of last computed centroids. (We note that the extra memory required by the alternative algorithms would be too demanding and would limit the scalability of our system to process millions of customers.)

3.2 Delta clustering

In our domain, new customers join the service, others discontinue the service, and still others make purchases, all on a daily basis. However, all of these are very low probability events. Hence, customer histories change very slowly.

Because of these slow changes, we can overlook the evolution in the customer base over short periods of time without any significant loss in precision. We can then re-assign (if necessary) every day those users with new purchasing activity in the previous day; we starting from the status of the latest execution and use the centroids found in the latest run as a starting point (after the new purchase data is collected). Cluster centroids, and a truly full clustering run, is only conducted every two weeks. This allows us to reduce considerably the number of iterations and thus the total execution time needed. The new clustering schema will include the recent users' activities, and depending on the purchasing of a specific content, a user may switch to a different cluster that shows a greater affinity with her new purchase history.

Keeping clusters with a stable population for longer periods of time provides also additional benefits: not only does it reduce computation time, it also reduces the likelihood of sending multiple exposures of the same message to a significant number of users (the negative effect of multiple exposures is discussed in Section 9.1). Indeed, when customers with different past viewing histories are re-grouped together, it becomes more difficult to satisfy the no-multiple-show condition. Also, frequently changing customers might lead the system to discard a good offer too frequently, just because a significant part of the cluster has seen it before.

Hence, in our application we made a trade-off between how often to do a complete re-clustering, and how long to maintain the population within each cluster stable. This is however an empirical question (we were able to define an adequate frequency for re-clustering after only few trials).

3.3 Number of clusters

Finally, choosing the number of clusters k is always a challenging task that depends on many factors such as customer base size and number of categories. In general, a larger number of clusters produces a more precise targeting. However, a larger number of clusters requires a longer clustering execution time and data preparation time, larger storage space, and a longer message delivery process. The latter one is a hard limit imposed by the carrier. Sending messages to all clusters is time consuming, as the delivery engine, for technical reasons, has to pause for few minutes between two consecutive deliveries. In addition, for marketing reasons, all customers have to receive messages only within a well-defined time frame. Hence, we need to make sure that the number of cluster is small enough not to extend the sending phase over such time frame.

The final choice on the number of clusters depends upon the available storage, computation power, and the gains that adding further clusters might provide in terms of predictive accuracy. In our empirical application we have consistently found that using about 20-30 clusters provides very good results. This is because, as it can be seen from Figure 1, performance improvements beyond an 11 cluster solution are minimal and beyond a 20 cluster solution are practically inexistent.

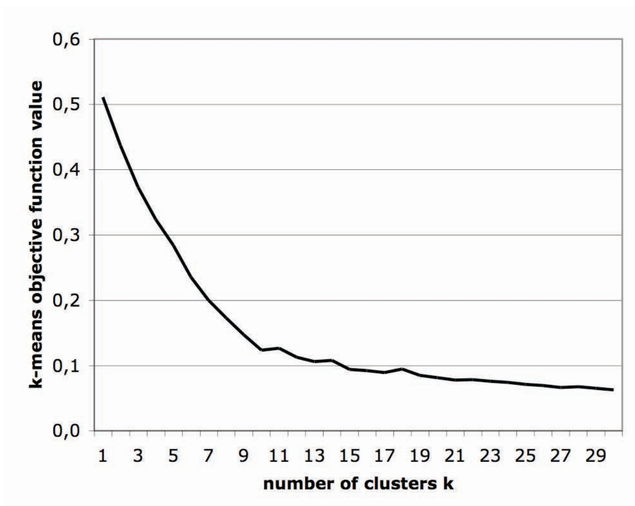

Figure 1. Clustering quality as a function on the number of clusters. The lower the value of the k-means objective function the better the overall clustering

3.4 Managing non-clickers

In our customer base we observed that only 35% of the population had purchased something in the past (*clickers*). We can only use the activity of these customers to derive the clustering schema, given that clusters are based on purchasing histories. For the remaining 65% of our customers (*non-clickers)* we have no historical information as they have never purchased anything.

To try to get usable information from *non-clickers*, we propose in our optimization system to send *good* offers to these customers (i.e., offers that tend to perform well overall). To identify these offers we compute offer performance among the entire clicker population (regardless of the clustering schema). In addition, in order to avoid pushing only few offers (given the system constraints mentioned previously that allow only to send one single message to a cluster or group of customers), we split the non-clickers group into smaller sets, each with about fifty thousand users. Then, we target each set of non-clickers using the category purchasing likelihood discussed previously. In other words, the probability that each set of non-clickers receives a specific content from a specific category is related to the purchasing probability of that category. By doing this we also reduce the risk of picking one bad offer and sending it to a large number of customers.

Finally, each new customer, upon arrival, is first inserted into these non-clickers sets. The customer will then be assigned to clicker groups (through full clustering or delta clustering) as soon a purchase is made.

4. LEARNING ON NEW OFFERS

Every day dozens of new offers are added to the catalogue and it is important to learn their purchasing likelihood as fast as possible (e.g., many of the offers will only be available for short periods of time, and some are associated to specific calendar events). The only information we have on new products is their category.

4.1 Category diversity

Category information could be highly valuable to infer the quality of new offers (in the absence of actual purchase histories from previous testing). However, the challenges here are (1) the extremely large library of offers (as compared to the learning occasions) that expands at a significant pace, and (2) the myriad of offer categorizations the different vendors give the mobile phone company under study.

To solve these problems we propose the use of a common and finer categorization of all offers. To obtain this categorization we have merged all categories from our original data into a single uniform schema. We used pattern matching and text mining techniques applied to the title and the category text to define this schema (see for example the Naïve Bayes Classifiers in [7]). To validate the use of these finer categories, we have tested for their predictive ability by attempting to predict the click-through-rate (CTR) of the different offers based on the new automatically constructed categories; we then compare the predictive performance with that obtained using the previous fragmented categorizations. Our results indicate a much better accuracy when using the new and finer categories.

Though this corresponds to a very interesting research problem in the data mining and knowledge discovery domains, it is not the main focus of this paper. Further details on how we built the new offer categorization are available from the authors upon request.

4.2 Heterogeneity within categories

Though categories are useful in predicting performance, it is very likely for different products/offers in the same category to show substantial differences in terms of purchasing probability.

Learning clusters

To deal with this problem we have allocated some users to what we call *learning clusters*. Hence, daily, the system selects a small random portion of our customer base and divides it into a set of learning clusters. Few thousand users compose each learning cluster. We note that we do not keep the learning clusters as fixed; these are created daily and can contain different users every time.

Producing the learning clusters daily might seem time consuming. However, we have conducted several experiments and monitored our optimization system and concluded that fixed learning clusters are not adequate in this context. Figure 2 presents an example of the experiments we conducted. In this figure we report the average CTR of a *fixed* learning cluster compared to optimized clusters (in *fixed* learning clusters, over the entire testing period, users are the same; optimized clusters are those cluster that receive optimized content).

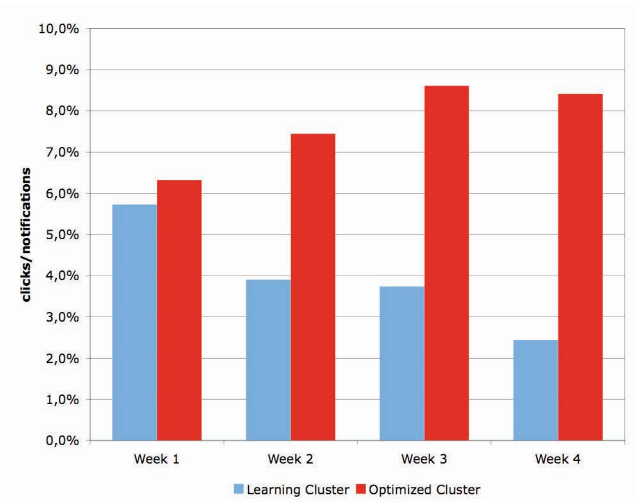

Figure 2. Fixed learning cluster vs. optimized cluster

As it can be seen from the figure, a typical optimized cluster has a good CTR, though it varies depending on the availability of good content (i.e., dependent on the quality of the contents available to send). In contrast, for the learning cluster, we observe a systematic decrease of CTR week after week. For week 1 the CTRs of both type of clusters are not significantly different (at 5% significance level). For the following weeks, the CTR differences are not only substantial, they are also statistically significant. This result suggests that customers might loose interest in the service if subject to prolonged exposure of bad content (i.e., content that is not targeted to the specific interests of the user). This is because the likelihood of receiving a bad content is high for a learning cluster (as the offers are not filtered based on previous learning). In fact, we find that for learning clusters the number of weak offers is higher than the number of good ones, given the total number of active offers in any moment. (We note that Figure 2 results also reveal that the optimization we propose can indeed provide substantial gains, a result that we will further discuss later in the paper.)

Hence, to prevent this problem we conclude that learning customers should indeed be *rotated*: learning clusters should be formed periodically with randomly assigned users. That is exactly what we do in our optimization system. Basically, learning clusters are formed by temporarily *borrowing* users from optimized clusters.

In order to monitor when such rotation might be required we could look at customer inactivity rate (i.e., the percentage of people that decide to stop downloading messages in the period under study), and at the rate of customer churn (i.e., the percentage of people that unsubscribe the service in the period under study). For example, during the four weeks of Figure 2, 3.8% of the customers unsubscribe the service for the learning cluster, against 1.6% for the revenue cluster. In addition, the learning clusters show an inactivity rate of 6.2% on average, versus 3.5% for the optimized clusters. This is part of further research we are currently conducting. For the current application, and to simplify its technical implementation, we randomly assign users to learning clusters every day, which is the minimum possible time period we can act on.

We note also that rotating users provides additional benefits. For example, by moving customers from optimized to learning we may discover customers' interests that could otherwise never be uncovered. In fact, in the learning clusters people are exposed to a greater variety of offer categories. Because customers' interests can change over time (e.g., shopping for a new car when having a baby, or looking for a mortgage when marrying), by keeping a customer in the same optimized cluster for a long time can lead the system to expose him/her to a limited number of categories and prevent the discovery of his/her new interests.

Heuristic to improve learning

The number of learning clusters used might depend on the number of new items on which we need to learn. For overall efficiency considerations, we cannot allocate more than a small portion of the entire customer base to learning because customers we allocate for learning do not produce optimized revenue. In addition, in the current application there are many offers with limited life-span (e.g., they are only valid for short periods of time which include seasonal offers associated to the holidays). Thus, it is very likely that we will not be able to learn fast enough on all items.

To alleviate this problem, in our optimization system we send the most recent new offers to learning clusters. In addition, we rank all categories based on overall performance, and we start learning on those items belonging to the most attractive categories. Finally, the system tries to mix content categories in each learning cluster. This is made in order to expose each learning cluster to a variety of topics.

4.3 Performance measures

To conduct all the activities described previously, we need to define a specific performance measure. In our application, we define the *potential* of an offer *n* as:

$$potential(n) = CTR(n) \cdot price(n).$$

where *price(n)* is the price of the offer *n* and *CTR*(*n*) is:

$$CTR(n) = clicks(n) \,/\, notifications(n)$$

with *clicks*(*n*) being the number of customers who purchased the content *n* and *notifications*(*n*) the number of customers who were exposed to that content. Basically, *potential*(*n*) measures the expected revenues for an exposure of a specific offer. (Our choice of content for each cluster is based on the potential as defined here.)

For certain contents, such as daily news or sport news, we cannot measure their potential as their life spans over a very short period of time. Hence, we tie their potential to the arrival time, that is, recent news have higher potential than older ones.

5. THE TARGETING ALGORITHM

An important component of our system is the selection algorithm that decides the offer to target to each cluster. This algorithm is based on the following considerations:

1) Cluster interests vary with respect to each category.
2) Using content *potential* instead of CTR is more appropriated in our application domain.

3) Reducing multiple exposure of same content to the same customer improves the overall performance (see our experiments reported in the Appendix).
4) Avoiding conflicts among offers sent together within the same MMS.
5) Contents which have been more recently seen influence customers' interests the most.
6) Contents order in the MMS has an impact on the CTR (please see the experiments we report in the Appendix).
7) Using a probabilistic selection algorithm reduce errors due to learning defects.

We now elaborate on each of the above considerations.

Points 1) and 2) are the basis of the targeted choice of content for each cluster. Basically, we tend to send the best offer (i.e., most likely to be purchased) to the most likely customers.

Point 3) reduces the probability of exposing the same customer multiple times to the same content. In the Appendix we discuss the effects of sending the same content more times to the same customers. For example, we find that after only a few exposures the CTR often decreases substantially (e.g., a decrease of about 50% is not uncommon).

Hence, every day the system computes the following value for each content-cluster pair:

$$views(n,C) = \sum_{u \in users(C)} \sum_{date \in seen(u,n)} w(today - date)$$

where *users*(C) denotes the set of customers belonging to cluster C; *seen*(u,n) is the set of dates on which user u has seen content n, and $w(today - date)$ is a weight function that gives greater weight to more recent impressions.

Basically, *views*(n,C) indicates *how much* the content n was seen in cluster C. This is based on the current population of C and when content n was last seen by each customer in C (for recently seen contents the function *views* increases).

We use the following *stop condition* to avoid multiple exposure of the same content to the same customers:

$$views(n,C) > size(C) \cdot threshold$$

where *size*(C) denotes the cardinality of cluster C. The *threshold* value ranges from zero to one. It allows us to stop sending a particular content to the cluster C if at least *threshold* percent of the people in C have already seen it. In particular, by setting *threshold* to zero we stop sending content n to cluster C as soon as one person in C receives n.

Point 4) derives from the consideration that usually customers do not buy more than one product on a single day from the same message. Since we try to reduce the probability of sending twice the same content to each user, it is safer to diversify as much as possible the categories we sent on a single MMS. For instance, if somebody is interested in wallpapers we try to send one wallpaper offer (that highly matches his/her interests) combined with products from other categories. By doing so, we are also able to target better people within the same cluster with slightly different interests.

Point 5) is somewhat an extension of the considerations done in 4). In this case we try to maximize category diversity over a longer period of time (days). That is, even though a customer is interested in a specific category, we try not to send only offers from that category over several consecutive days. Again, this has two benefits. One, by exposing customers to more categories we can learn their potential interests towards alternative content. Two, we prevent customer boredom that could be caused by showing only few content categories.

Point 6) is based on the outcome of the experiment discussed in Section 9.2. That is, the order used to show contents with a single message impacts the overall CTR. As shown in the Appendix, the CTR measured on the first position is in general twice as much as the one in position two and three. Thus, we show contents in a message based on their potential, by showing in the first position the content with highest potential and so on (this way we can maximize potential).

The considerations on 7) are aimed to compensate possible learning defects by selecting offers from each category on a probabilistic basis rather than on a deterministic one. That is, each category has a chance to be selected proportional to its potential.

Learning defects could be related to different factors such as:

- the reduced sample size we use for learning (5 to 7K);
- noise in the data (clicks and notifications) received from the carrier (even though our system tries to detect anomalies in the data [10], it not always succeeds);

A simplified version of our algorithm is depicted in the following:

```
for C in clusters:
        selected=new list()
        while selected.length() < num_contents:
                for P in C.preferredCategories():
                        n = P.selectBest(C.seen()
                                U selected)
                        if n!=null:
                                selected.append(n)

        to_send = extract(selected,num_contents)
        camp = new campaign(to_send)
        system.sendMMS(C,camp)
```

`C.preferredCategories()`: Returns a sorted list of categories for cluster `C` (from the most to least interesting). This is based on a probabilistic choice.

`P.selectBest(C.seen() U selected)`: Returns the best contents for category `P`. It excludes previously sent contents for cluster `C` (parameter `C.seen()`) and contents selected in previous loops (`selected`). Such selection is based on the content potential as previously discussed.

`extract(selected,num_contents)`: Extracts the first `num_contents` contents. It tries to select such contents from different categories. However, it may select from same categories in case it is not able to pick enough contents from different categories.

`new campaign(to_send)`: Creates a campaign composed of the contents contained in vector `to_send`. It sorts such contents based on the considerations above discussed.

`system.sendMMS(C,camp)`: Sends the campaign `camp` to cluster `C`.

6. OVERVIEW OF THE BASIC APPROACH AND SYSTEM WORKFLOW

Our approach is characterized by three building blocks: user clustering, performance learning, and message targeting. (Each one of these building blocks has been described in detail in the previous sections.) Hence, every day, the system we developed performs the following five steps:

I. *Data gathering and cleaning*: the database is updated with new data.

II. *User clustering*: Customer base is clustered based on all available data as described.

III. *Computation of cluster- and offer-specific statistics*: Summary statistics are computed for (1) cluster affinity towards categories, (2) generic category potential, (3) contents seen by each cluster, and (4) content potential.

IV. *Campaign scheduling*: The decision algorithm (discussed in Section 5) chooses the contents to be sent to each cluster and creates the related campaign. In a similar way, the system schedules recently added contents, that still needs to be learned, for the *learning clusters* (see Section 4 for details.)

V. *Sending*: Eventually, schedules on campaigns and related customer groups are communicated to the MMS sending platform for final delivery to mobile phones. This schedule specifies for each customer the set of offers to send on that day.

6.1 Data flow

Every day the targeting system collects information on new customers and new contents from the database carrier. It also collects statistics related to message delivery (e.g., notifications, purchases) from the message sending platform. We then update our internal database with this new information.

We then export to the message sending platform data needed for delivery (SMIL files and related MSISDN list).

Data flow is depicted in Figure 3 below. (Note that in our empirical applications messages were all of the multimedia type, i.e., MMSs.)

6.2 System architecture

Our system runs on a Linux platform. All software is written in Python [15], a high level object oriented programming language. It is well suited to manage flows and to make data analysis. All data are stored on IBM-DB2 [9], a DBMS that provides high scalability and robustness. The system is redundant by mirroring the database on a second server ready to take over in case of fault of the main server.

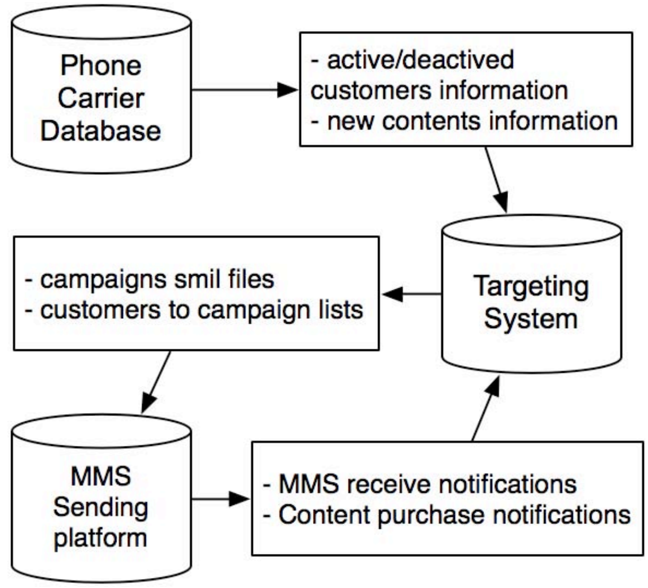

Figure 3. Data flow representation

7. RESULTS

Our system has been running successfully for more than a year in a real business environment. The customer base counts over two million customers and results show a considerable improvement compared to a non-optimized solution (during the first months we only collect learning data; then, we started clustering customers using these data).

We were not allowed to set up a control panel, which would have been ideal for testing our system. Thus, we performed our test by measuring overall performance before and after the usage of our optimization. We carried on the test over a ten-week period, five weeks before the activation of our system and five weeks after. We did not consider holidays in those days in order to make sure the two five-week periods are consistent. Hence, for the first five weeks we kept the optimization off. Thus, the system was sending messages based upon the carrier proprietary methodology.[1] For the following five weeks we switched on our optimization.

In Figure 4 we show the revenue improvement due to our system. Such revenue improvement is computed by dividing the overall revenue generated in a specified period by the number of messages delivered in that period. Thus, it denotes the normalized currency value per notification (for confidentiality reasons we cannot disclose actual revenue values).

Results show a significant benefit from using the optimization we propose. The average value computed during the first five week is 0.07 whereas the one computed over the optimization time is 0.16. That means an improvement of 141% over the performance of the proprietary algorithms (our baseline). This is a significant improvement. Indeed, one year after the introduction of our system, management perception was that of a substantial

[1] Details on the methodology used previously by carrier have not been revealed to us. However, knowing the exact algorithm previously used is irrelevant for the validity of our comparison. We simply used that period as benchmark for our system.

improvement in overall business performance with substantial increases in revenue.

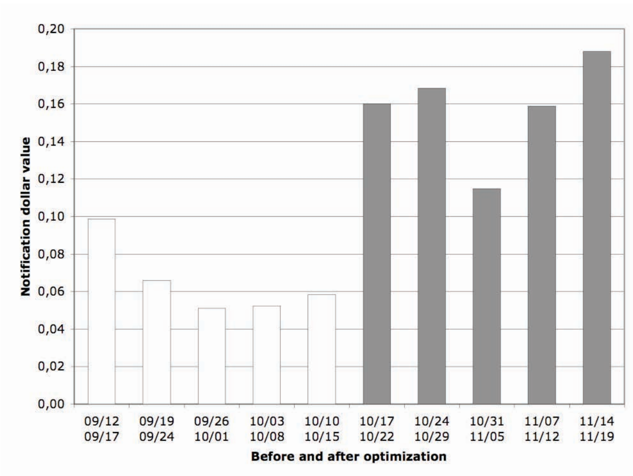

Figure 4. CTR before and after clustering

8. FUTURE RESEARCH

Though the clustering and targeting system we propose has provided a significant performance increase, we are currently considering several possible improvements. Learning on dynamic environments like the one of our application presents many challenges. For example, many of the offers can have an incredibly limited life-span (e.g., few days or even few hours). In such cases we are required to learn extremely fast. However, it often occurs that the type of learning we use in this application is not feasible because the total learning time is longer than the content life-span. A related problem results from the fast arrival rate of new catalogue items. Learning on all of those new items usually requires a very large learning space (e.g., through the assignment of more customers to learning clusters). The problem of using a large learning space is that we can considerably reduce the system's overall performance because we reduce the optimization space.

To alleviate these problems we may need to learn on some common characteristics. In [2] we presented some experimental results where we combine text mining on the offer's text with the visual characteristics of the associated image. The initial results were encouraging but more work still needs to be developed in this area.

Other future research might also include work aimed at improving the generalizability of the targeting algorithm we propose here. We feel we need to experiment more on users' behavior to find other indicators to be included in our algorithm. In addition, we also intend to develop more refined buying behavior models. For example, it would be interesting to understand the impact of today's purchase on the purchase decisions tomorrow (in other words, how spending some money today affects customers' purchasing decision on the following days). We believe that understanding and incorporating such type of models in our algorithm may definitively improve overall performance.

Another interesting future research path would be to understand the best time of the day (and day of the week) to send a promotional message to each user. At this time we did not include any temporal consideration in our algorithm. MMS messages are currently sent at the same time to all customers, though it could be interesting to understand whether the time of the day influences the purchasing probability. In a similar manner, we could embed (if available) location-based information into the recommendation engine. This would open up interesting research avenues as people may be treated differently depending upon their current geographical location at the moment the SMS/MMS is sent.

9. APPENDIX - EXPERIMENTS ON THE CUSTOMER BASE

We have conducted several experiments to determine how the algorithm structure might influence the *Click-through rate* (CTR) of each offer. In all the experiments we used random samples of about 11 to 12 thousand mobile-phone users. The content being tested in these experiments had never been sent to the users, and no information regarding its effectiveness was available. In addition, the alternative offers were equally priced, allowing us to ignore the costing factor.

9.1 Multiple Sending

Previous research seems to suggest that the number of exposures to a commercial message (e.g., a banner in the online) can have an influence on consumer response. For example in [4] the authors find that repeated banner exposures can increase the CTR rate. Because the medium we are exploring lacks sufficient research in these areas, we have conducted a series of experiments to determine the relationship between offer exposure and clicks. Our goal is to understand how the CTR of a single offer changes with the number of exposures. To do so, we sent repeated exposures of the same content (e.g., content A) to a random sample of users over a period of 10 days. In the example below you can see the results for a test in which the content was sent every three days. During the remaining days users were exposed to other offers (for a total of seven different offers; for example content *A*, *B*, *C*, *D*, *E*, *F*, and *G*). Only one offer (offer A) was sent multiple times during these testing days (in this example the final pattern of exposure was **A** – B – C – **A** – D – E – **A** – F – G – **A)**.

Figure 5 presents the CTR of each one of the offers sent during the 10 consecutive days under study.

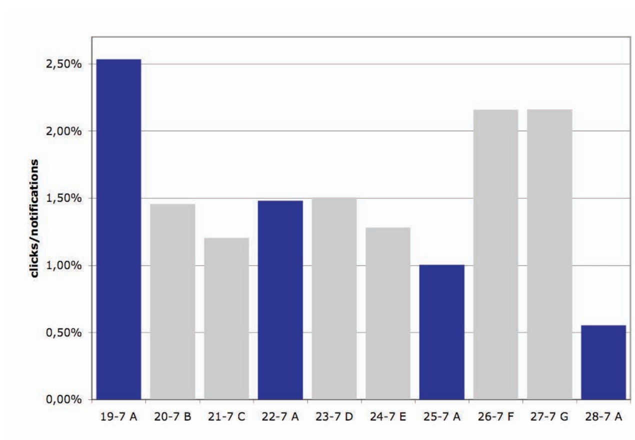

Figure 5. Click-Through-Rate of Seven Offers Sent Over Ten Consecutive Days (from 19/07 till 28/07)

The results across these experiments clearly show a significant decrease in CTR of a given content as the number of exposures increases. This is indicative of reduced interest in the content, that is, the content has been seen but not immediately clicked on. For example, in the example above, after the first exposure, the CTR of the second exposure is about 42% lower than the CTR of the first exposure. The CTR of the third exposure is also significantly lower: about 60% lower than the first exposure CTR.

9.2 Offer position in a message

Previous research suggests that content order has a significant impact on CTR [1]. In a second set of experiments we study how changing the offer's position in a message influences the final CTR of the offers in our mobile phone environment. To do so, in these experiments we drew three groups of random customers (G_1, G_2 and G_3) and randomly select three offers (content *A*, *B* and *C*). We then sent to each group the same three offers in a single message, but changed the order in which the contents would appear on the users' handsets. We sent the contents in the following order: (A, B, C) to group G_1, (C, A, B) to group G_2, and (B, C, A) to group G_3.

We computed the average CTR for each position and across the different offers. We present the results of one of these experiments in Figure 6.

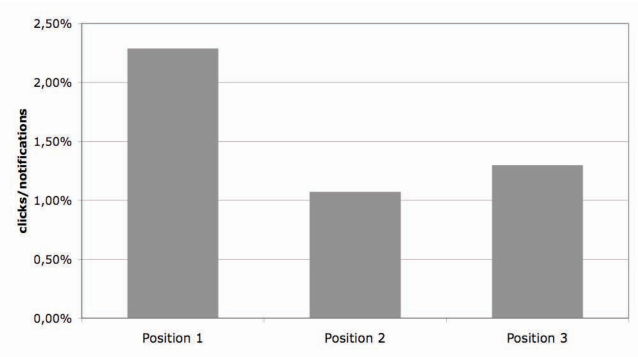

Figure 6. Click-Through-Rate of Contents at Different Positions

Figure 6 clearly reveals that content sent in the first position is two times more likely to be effective than content sent in the second and third positions (the difference between the second and third position is not statistically significant at 5% significance level).

Another important issue, regarding message optimization, would be to determine how to position the alternative offers in the message. Each offer can have significantly different levels of attractiveness (as measured by CTR). For example, in this experiment, we ranked the contents based on user response. On average users click on content *A* more often. Content *C* is the second best, followed by content *B*, the offer with the lowest CTR. In addition, it is possible that the CTR of each offer might interact with its position in the message. If such interaction occurs, any message optimization will need to take into account not only overall CTR, but also the best position in a message given the expected CTR.

Figure 7 provides a clear answer on whether CTR and content position in a message do interact. For example, content *A*, which is the best among all three offers, performs the best when positioned first in the message. The difference in performance is so substantial that makes the combination with offer *A* positioned first in the message the best performing message. Indeed, for this experiment the best combination is (*A*, *B*, *C*), that is the message with the best content in the best performing position (first in the message), the second best content (content *C*) in the second best position (third in the message), and the weakest content (content *B*) in the worst position (second in the message).

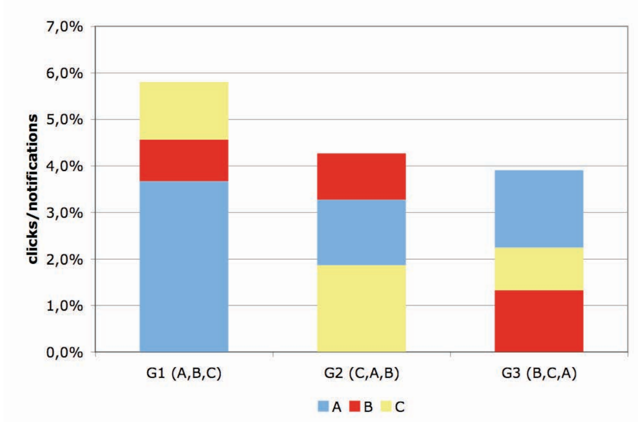

Figure 7. CTR by customer group

REFERENCES

[1] Ansari, Asim and Mela, Carl F. E-Customization. Journal of Marketing Research, 2003, 40(May), 131–145.

[2] Battiato, S., Farinella, G. M., Giuffrida, G. and Tribulato, G., Data Mining Learning Bootstrap Through Semantic Thumbnail Analysis, *Proceedings of SPIE Electronic Imaging 2007*, San Josè (CA, USA), 2007.

[3] Bauer, H. H., Neumann, M. and Reichardt, T., Driving consumer acceptance of Mobile Marketing - A theoretical framework and empirical study. *Proceedings of the 4th International Marketing Trends Congress, ESCP-EAP Annual Conference.*, Paris, 2005, 181–192.

[4] Chatterjee, Patrali, Hoffman, Donna L., and Novak, Thomas P. Modeling the Clickstream: Implications for Web-Based Advertising Efforts. *Marketing Science*, 2003, 22(4) 520–541.

[5] Dhillon, I. S. and Modha, D. S. Concept decompositions for large sparse text data using clustering. *Machine Learning*, 2001, 42(1):143–175. Also appears as *IBM Research Report RJ 10147*, 1999.

[6] Dhillon, I. S., Fan, J. and Guan, Y. Efficient clustering of very large document collections. *In Data Mining for Scientific and Engineering Applications*, 2001, 357–381.

[7] Domingos, P. and Pazzani, M. On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning*, *29*,1997,103–130.

[8] Fennell, Geraldine, Allenby, Greg M., Yang, Sha and Edwards, Yancy. The Effectiveness of Demographic and Psychographic Variables for Explaining Brand and Product Category Use. *Quantitative Marketing and Economics*, 1(2), 2003, 223–244.

[9] IBM DB2 http://www.ibm.com/db2/

[10] Iglewicz, B. and Hoaglin, D.C., How to detect and handle outliers. *ASQC Quality Press*, Milwaukee (Wisconsin), 1993.

[11] Kolda, T. G.. Limited-Memory Matrix Methods with Applications. PhD thesis, *The Applied Mathematics Program*, University of Maryland, College Park, Maryland, 1997.

[12] Mac Queen, J.B. Some methods for the classification and analysis of multivariate observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 1967, 281–297.

[13] Montgomery, Alan L. Using Clickstream Data to Predict WWW Usage. Working paper available at http://www.andrew.cmu.edu/user/alm3/papers/predicting%20www%20usage.pdf [last accessed: 16/09/07], 1999.

[14] Netsize, Convergence: Everything is going mobile. *The Netsize Guide 2007*, ISBN: 2-9523533-2-8, http://www.netsize.com/, 2007

[15] Python Programming Language, http://www.python.org/